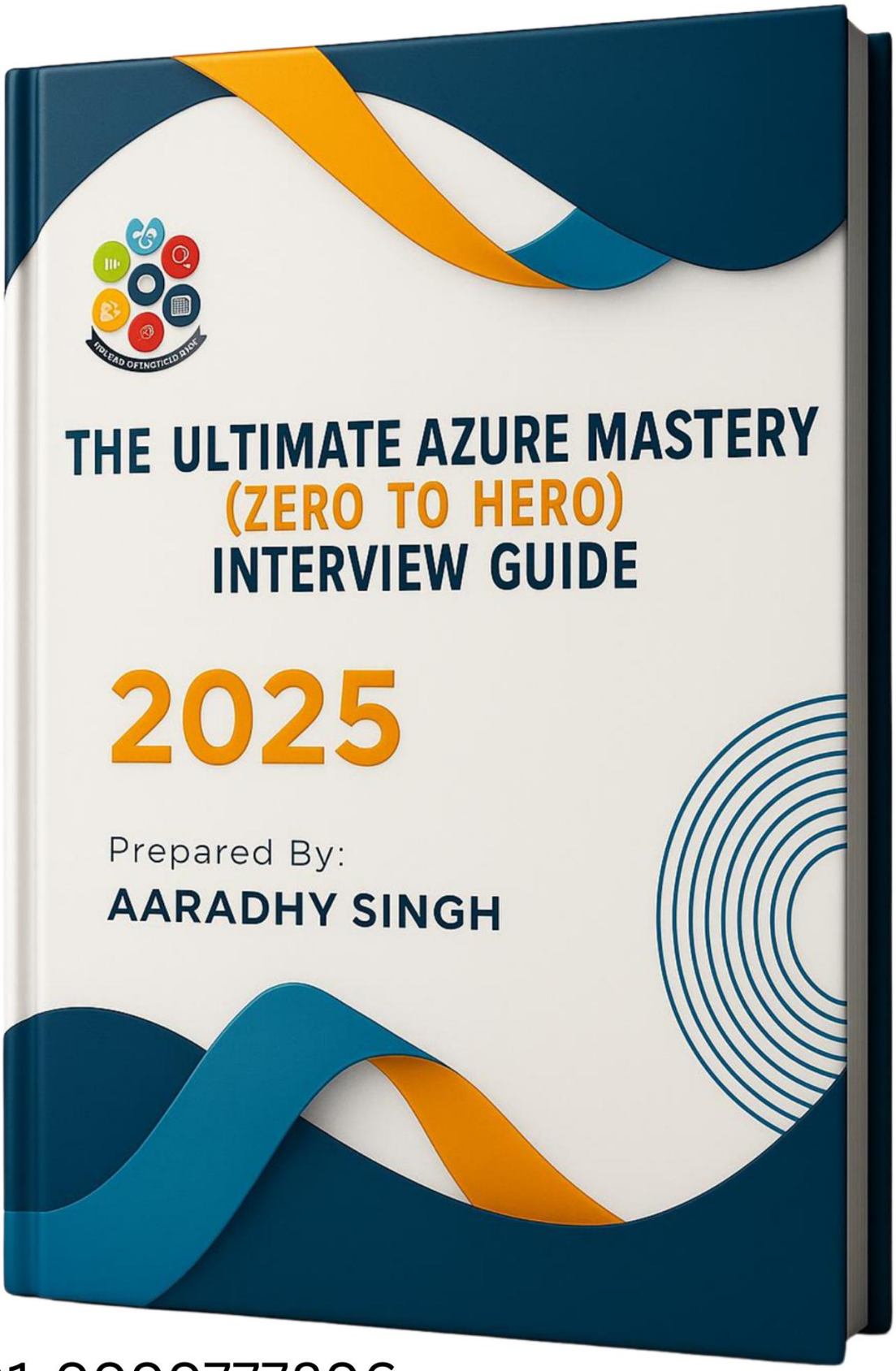


+91-9999777896

https://www.linkedin.com/in/aaradhy-singh-46bb0b26/

Contents

1. Azure Storage Account Overview

Azure Storage Account is a Microsoft Azure PAAS service that provides **highly available**, **secure**, **scalable**, and **durable** storage for a wide variety of data objects in the cloud.
It acts as a **container** for multiple types of data storage services.

Types of Azure Storage Service in Storage account

Storage Type	Description	Common Use Cases
Blob Storage	Object storage for unstructured data like images, videos, backups, and logs	• Store documents, images, videos • Host static websites • Big data analytics (data lakes) • Application logs and backups
File Storage (Azure Files)	Managed file shares accessible via SMB or NFS protocol	• File share for lift-and-shift Windows apps • Shared config files for distributed apps • Map network drives from on-prem to cloud
Queue Storage	Message storage for asynchronous communication between apps	• Decouple services in distributed applications • Process background jobs or workflows • Improve reliability via message queuing
Table Storage	NoSQL key-value storage for structured datasets	• Store large sets of structured, non-relational data • Ideal for telemetry, user profiles, product catalogs, device metadata, logs
Data Lake Storage	Optimized for analytics workloads (batch, streaming, AI/ML). Provides hierarchical namespace (like folders & directories, unlike normal Blob flat structure). Fully integrated with Azure services like Databricks, Synapse, HDInsight, Power BI.	• Big data analytics • IOT, Streaming data • Data warehousing

Types of Azure Storage Accounts

Account Type	Features	Best For
General-purpose v2	Supports all storage types (blob, file, queue, table, disk) + latest features	Most common choice, supports all workloads
General-purpose v1	Legacy, fewer features, cheaper	Avoid unless for legacy compatibility
Blob Storage Account	Optimized specifically for blob storage with hot/cool/archive tiers	Applications needing object storage
Block Blob Storage	Premium performance for block blobs (SSD-backed)	High-performance needs like media rendering
File Storage	Premium file shares using SSD	Enterprise-grade applications needing fast IO

Key Features

- **Redundancy Options:** LRS, ZRS, GRS, RA-GRS
- **Access Tiers:** Hot, Cool, Archive (for Blob)
- **Secure:** Supports encryption, private endpoints, firewalls
- **Scalable:** Can handle exabytes of data and millions of requests

Azure Storage Redundancy Options

Redundancy ensures your data is **durable** and **available**, even in case of hardware, zone, or region failures.

1. Locally Redundant Storage (LRS)

Redundancy Scope: Single datacenter (within one Azure region)
Copies: 3 synchronous copies
Cost: Lowest
Durability: Protects against local drive/rack failure
Use Cases:

- Dev/test environments
- Non-critical backups
- Regulatory restrictions requiring data to stay in one region

2. Zone-Redundant Storage (ZRS)

Redundancy Scope: Multiple Availability Zones in a single region

Copies: 3 copies across physically separated zones

High Availability: Protects against datacenter or zone failures

Supported for: Blob, File, Disk, and Queue (in supported regions)

Use Cases:

- Production applications needing high availability
- File shares for business-critical apps
- Data lakes or blob storage for resilient data pipelines

3. Geo-Redundant Storage (GRS)

Redundancy Scope: Primary + secondary **paired Azure region**

Copies: 3 local (LRS) + 3 geo (asynchronous)

Data replicated across 400+ miles

Use Cases:

- Disaster recovery (DR) planning
- Critical business data like invoices, reports
- Compliance requirements with geo-redundancy

4. Geo-Zone-Redundant Storage (GZRS)

Redundancy Scope: Combines **ZRS + GRS**

Copies: Zone-redundant in primary region + geo-redundant in secondary

Best of both worlds: High availability + DR

Use Cases:

- Enterprise workloads with strict HA + DR needs
- Financial apps, healthcare, government data
- Data lakes with critical operational impact

5. Read-Access Geo-Redundant Storage (RA-GRS)

Redundancy Scope: Same as GRS

Plus: Read access to secondary region during outages

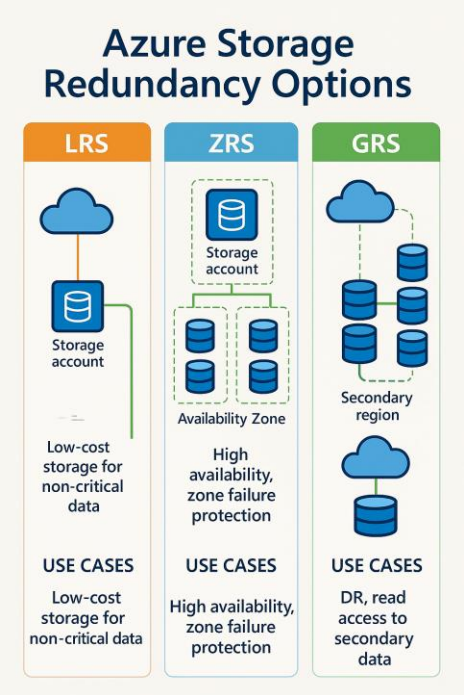
Use Case Bonus: Read data during region outage or failover test

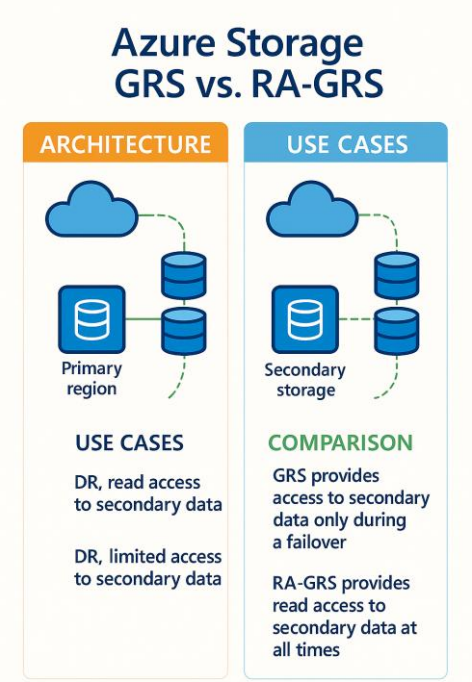
Use Cases:

- Reporting workloads in DR regions
- Audit, compliance systems
- Read-heavy analytics across regions

Summary Table

Redundancy Type	Zones	Regions	Copies	Use Case Keywords
LRS	❌	❌	3	Low cost, basic
ZRS	✅	❌	3	High availability
GRS	❌	✅	6	DR, compliance
GZRS	✅	✅	6	HA + DR critical
RA-GRS	❌	✅	6	Read access in DR





1- Blob Storage

Azure Blob Storage is Microsoft's **object storage solution** in the cloud.

- "Blob" = **Binary Large Object** (like files, images, videos, logs, backups).
- It is designed to store **massive amounts of unstructured data** (data that doesn't follow a fixed schema).

Access Tiers

Azure Blob Storage offers **three main access tiers** that help you manage your storage cost based on how frequently your data is accessed.

Tier	Frequency of Access	Storage Cost 💰	Access Cost 📥	Minimum Retention
Hot	Frequent	High	Low	None
Cool	Infrequent (≥30 days)	Lower	Higher	30 days
Archive	Rare (≥180 days)	Lowest	Highest	180 days

1. Hot Tier

- **Purpose:** For data that is accessed or modified frequently.
- **Fast read/write** performance.
- No minimum retention period.

Use Cases:

- Web/mobile app content
- Active logs and backups
- Frequently accessed media files
- Real-time analytics

2. Cool Tier

- **Purpose:** For data accessed less frequently but still needs quick availability.
- Lower storage costs but higher access costs.
- Minimum 30-day retention.

Use Cases:

- Monthly backups
- Older media files
- Archived customer invoices (for 2–3 months)

- Email or document archives

3. Archive Tier

- **Purpose:** For rarely accessed data that can tolerate **hours of retrieval time**.
- Cheapest storage, but **expensive and slow** to access.
- Minimum 180-day retention.

Use Cases:

- Long-term compliance (financial/medical records)
- Historical data (IoT logs, audit logs)
- Regulatory backups
- Archived photos, videos, scientific datasets

Tier Lifecycle Management

Azure allows **automatic movement** between tiers using **lifecycle management policies**, such as:

- Move blobs to cool tier after 30 days of inactivity
- Move blobs to archive after 180 days
- Delete blobs after 1 year

Summary Cheat Sheet

Use Case	Suggested Tier
Real-time logs and app data	Hot
Monthly report backups	Cool
Legal compliance data (7+ years)	Archive
Streaming video (frequently used)	Hot
Archived security footage	Archive

2- Azure File Storage

Azure File Storage provides **fully managed, cloud-based file shares** that can be accessed over **SMB (Server Message Block)** or **NFS (Network File System)** protocols. It allows **multiple users or applications** to access the same files **concurrently**, just like a traditional file server.

Key Features

Feature	Description
SMB & NFS Support	Access file shares using SMB 3.0 (Windows/Linux/macOS) or NFS v4.1
Fully Managed	Microsoft manages hardware, patching, and availability
Mountable Anywhere	Mount Azure file shares to VMs, on-prem servers, or containers
Hybrid Access	Combine with **Azure File Sync** to cache cloud files on on-prem servers
Snapshot Support	Take point-in-time snapshots for recovery
Access Control	Supports RBAC, NTFS ACLs, and AD authentication (via Azure AD DS)
Redundancy Options	Supports LRS, ZRS, GRS, and RA-GRS for high durability

File Storage Tiers

Tier	Use Case	Performance	Backed By
Standard	General-purpose file shares	HDD-backed	Hard Disks
Premium	High-performance workloads	SSD-backed	SSD
Transaction-Optimized	Medium performance & high operations	Optimized for metadata-heavy workloads	HDD
Hot / Cool	Based on usage frequency	Cost-optimized	HDD

Storage Types

Type	Description
Standard File Shares	Pay-as-you-go capacity and performance
Premium File Shares	High IOPS and throughput (dedicated SSD)

Azure File Storage Use Cases

Use Case	Why File Storage Works Well

Lift-and-shift apps	Migrate file-dependent Windows apps to the cloud
Shared file access	Share files between multiple VMs or containers
User profile storage	Store FSLogix profiles in Windows Virtual Desktop
Backup and restore	Snapshot support for quick file recovery
Hybrid file server replacement	Use Azure File Sync to extend on-prem file server
Configuration & log storage	Central file share for app configs or logs

Azure File Sync (Hybrid Scenario)

Azure File Sync allows you to:

- Cache **hot files on-prem** and store **cold files in the cloud**
- Enable **multi-site access**
- **Sync file changes bi-directionally**

Great for organizations migrating to the cloud gradually.

Pricing Considerations

You pay for:

- **Provisioned storage (GB/month)**
- **Operations (per 10,000 calls)**: Read, write, list, snapshot, etc.
- **Data transfer**: Outbound data beyond Azure region

Azure File Storage vs Blob Storage

Feature	**Azure File Storage**	**Azure Blob Storage**
Protocol Support	SMB, NFS	REST/HTTPS
Access	Mount as drive or share	Access via APIs or tools
Ideal For	File servers, shared access	Media, backups, data lakes
Directory structure	True file system	Flat namespace (folder simulated)
AD/ACL integration	Yes (NTFS, Azure AD DS)	Limited

Best Practices

- Use **Premium tier** for high-performance workloads
- Use **File Sync** to connect on-prem file servers to the cloud
- Use **Snapshots** for quick file-level recovery and Restrict access using **RBAC**, **ACLs**, and **network rule**

3- Queue Storage

Azure Queue Storage is a **message queue service** for storing large numbers of messages that can be accessed from anywhere via authenticated HTTP or HTTPS calls.

It allows asynchronous **communication between application components** to ensure scalability and resilience.

Key Features

Feature	Description
Message Queueing	Stores and retrieves messages asynchronously
Durable Messaging	Messages are stored until processed or expired
High Scalability	Can handle **millions of messages per queue**
Time-to-Live (TTL)	Messages can have an expiration time (max: 7 days)
REST API + SDK support	Accessible via HTTP calls or SDKs (.NET, Java, Python, etc.)
Poison Message Handling	Manual logic to handle retry-failed messages

How It Works

- You **enqueue** a message (max size: 64 KB).
- A backend service **dequeues** and processes it.
- Messages are invisible to others while being processed.
- You can **update**, **delete**, or **re-queue** if needed.

Use Cases

Use Case	Why Queue Storage Works Well
Asynchronous task processing	Decouples front-end from background processors
Order processing systems	Processes transactions in a reliable sequence
Background jobs / Batch processing	Offloads heavy tasks from main app thread
Email/SMS notifications	Queue messages for scheduled or retry delivery
IoT data ingestion	Buffer telemetry before processing
Workflow management	Chain tasks and trigger next steps asynchronously
Retry logic / Delay handling	Use visibility timeout to reprocess failed tasks

Pricing Highlights

- Charged per:
    - **Storage** used (GB/month)
    - **Operations** (enqueue, dequeue, peek, delete)
- Very **cost-effective** for simple queuing

Example: Message Lifecycle

1. **Add Message (Enqueue)**
   → Frontend app places task {"OrderId":123, "Action":"Ship"}
2. **Retrieve Message (Dequeue)**
   → Worker reads and temporarily hides it (visibility timeout)
3. **Process Message**
   → Worker completes the job
4. **Delete Message**
   → If success, remove from queue. If failed, it becomes visible again.

Message Example (Base64 or Text)

{ "OrderID": "56789", "Action": "SendConfirmationEmail" }

Azure Queue vs Service Bus Queue

Feature	**Azure Queue Storage**	**Azure Service Bus Queue**
Protocol	HTTP(S)	AMQP / HTTP(S)
FIFO support	No (default), manual logic	Yes (via sessions)
Message size	Up to 64 KB	Up to 256 KB or 1 MB
Advanced features	Basic retry, TTL	Dead-lettering, sessions, filters
Cost	Very low	Moderate

Best Practices

- Use **visibility timeout** to avoid duplicate processing
- Set TTL to automatically expire unprocessed messages
- Store large payloads in Blob, reference via message

4- Table Storage

Azure Table Storage is a **NoSQL key-value** store provided by Microsoft Azure, designed to store **large volumes of structured, non-relational data**. It offers fast and cost-effective access to data.

Key Features

Feature	Description
NoSQL key-value store	Stores data as entities with properties — no fixed schema required
Scalable	Can store **billions of rows** with fast access
Partitioned	Optimized for **high throughput** via partitioning
Low-cost	Very **affordable** for storing large datasets
Accessible via REST API	Use HTTP, SDKs (.NET, Java, Python, etc.)
Supports OData protocol	Easily queryable via HTTP and OData filters

Data Model Structure

Each Table Storage entity is a row with:

Field	Description
Partition Key	Groups entities for load balancing
Row Key	Unique ID within the partition
Timestamp	Auto-generated time of last update
Properties	Up to 252 custom key-value pairs

Use Cases

Use Case	Why Table Storage Fits
User profiles	Scalable key-value store, fast lookups
Audit logs / activity logs	Large volume of append-only data
IoT device data	Time-series, structured but dynamic schema
Product catalogs	Easily stores and retrieves items
Sensor/telemetry data	Ingests large, high-speed input cheaply

App diagnostics or metadata	Quick storage of small config values
Chat/message history	Lightweight, flexible data structure

Pricing Highlights

- **Charged per GB stored per month**
- Transaction cost depends on operation type:
  - **Insert, Update, Delete**: write operations
  - **Query**: read operations
- Very **cost-effective** compared to traditional RDBMS or Cosmos DB (in some cases)

Comparison: Table Storage vs. Cosmos DB Table API

Feature	**Azure Table Storage**	**Cosmos DB Table API**
Latency	Lower	Sub-10 ms reads/writes
Global replication	Manual	Built-in
SLAs	99.9%	99.999% (with multi-region)
Throughput	Limited per partition	Highly scalable
Pricing	Very low-cost	Premium

5- Data Lake Storage

Azure Data Lake Storage (ADLS) is a highly scalable and secure data lake service built on top of Azure Blob Storage. It is designed to handle big data analytics workloads by storing structured, semi-structured, and unstructured data at virtually unlimited scale.

It comes in **two generations**:

- **ADLS Gen1** (legacy, being retired)
- **ADLS Gen2** (current, recommended) → built on Blob Storage with **Hierarchical Namespace (HNS)**

Key Features of ADLS

1. **Scalability**
   - Stores **petabytes to exabytes** of data.
   - Suitable for batch, streaming, and interactive analytics.
2. **Hierarchical Namespace** (Gen2 feature)
   - Organizes data in directories and files (like a file system).
   - Enables **atomic operations** (rename, move, delete).
3. **Cost Optimization**
   - Built on **Azure Blob Storage pricing model** (hot, cool, archive tiers).
   - Pay only for what you store and process.
4. **Security & Access Control**
   - **Azure Active Directory (AAD) integration** for authentication.
   - **RBAC** and **POSIX-style ACLs** for fine-grained access control.
   - Supports encryption at rest and in transit.
5. **High Throughput & Performance**
   - Optimized for **parallel data processing frameworks** (Apache Spark, Hadoop).
   - Works seamlessly with **Azure Databricks, Synapse Analytics, HDInsight**.
6. **Integration with Azure Ecosystem**
   - Can be used as a **data lakehouse** foundation.
   - Connects with **Power BI, Azure ML, Event Hub, IoT Hub, Data Factory**.
7. **Multiple Data Formats Supported**
   - CSV, JSON, Parquet, Avro, ORC, Images, Video, Logs, IoT data, etc.

Common Use Cases of ADLS

1. **Big Data Analytics, Data Lakehouse**
   - Store raw, semi-processed, and curated data.
   - Run analytics with Databricks, HDInsight, or Synapse.
   - Combine data lake flexibility + data warehouse performance.
   - Store structured + unstructured data, query with Synapse or Databricks.
2. **IoT & Streaming Data Storage**
   - Store high-velocity sensor, telemetry, and log data from **Event Hub / IoT Hub**.
3. **ETL / ELT Pipelines**
   - Use as a **staging area** for data ingestion.
   - Transform data with Data Factory, Databricks, or Synapse.

2. Azure Virtual Machine

1. What is an Azure Virtual Machine?

A **computing resource in Azure (IaaS)** that lets you run Windows/Linux OS and applications with full control.

2. What VM series are available in Azure?

- **A-series** → Entry-level, dev/test.
- **D-series** → General purpose.
- **E-series** → Memory optimized.
- **F-series** → Compute optimized.
- **L-series** → Storage optimized.
- **N-series** → GPU workloads.
- **H-series** → High-performance computing.

3. What is the difference between Managed and Unmanaged Disks?

- **Managed Disks** → Azure handles storage account, replication, scaling.
- **Unmanaged Disks** → User manages storage account (legacy).

4. What is Availability Sets in Azure VMs?

A logical grouping to distribute VMs across **fault domains (power/rack)** and **update domains (patching cycles)** to ensure HA.

5. What are Availability Zones?

Physically separate datacenters in an Azure region that provide fault isolation and redundancy.

6. What types of Azure disks are available for VMs?

- Standard HDD
- Standard SSD
- Premium SSD
- Ultra Disk

7. What is ephemeral OS disk in Azure VM?

A disk type stored on local SSD, ideal for **stateless workloads** – faster and cheaper but not persistent.

8. How do VMs connect to a network in Azure?

Through **NICs connected to VNets** with support for NSGs, public IP, private IP, load balancers.

9. How do you secure RDP/SSH access to VMs?

- Use **Azure Bastion**.
- Enable **Just-in-Time (JIT)** access.
- Restrict inbound ports using NSGs.

10. What is Accelerated Networking?
A feature that enables **SR-IOV (Single Root I/O Virtualization)** to improve network throughput and reduce latency.

11. How are Azure VM disks encrypted?

- **Azure Disk Encryption (ADE)** using BitLocker/DM-Crypt.
- **Server-Side Encryption (SSE)** with Microsoft-managed or customer-managed keys.

12. How do you back up Azure VMs?
Using **Azure Backup** with Recovery Services Vaults.

13. What is Azure Spot VM?
Low-cost VMs that use **unused Azure capacity**, but can be evicted anytime.

14. What is Azure Reserved VM Instance (RIs)?
Prepaid VM commitment (1 or 3 years) at a **discount (up to 72%)** vs pay-as-you-go.

15. How do you automate VM creation?
Using **ARM templates, Bicep, Terraform, Azure CLI, or PowerShell**.

16. What's the difference between VM Scale Sets (VMSS) and individual VMs?

- **VM** → Single compute resource.
- **VMSS** → Auto-scaling group of identical VMs for elasticity.

17. How do you monitor Azure VMs?

- **Azure Monitor** (CPU, memory, disk, network).
- **Application Insights** (app-level monitoring).
- **Log Analytics** (query-based insights).

18. How do you implement disaster recovery for Azure VMs?

Using **Azure Site Recovery (ASR)** to replicate and failover VMs to a secondary region.

19. What is a proximity placement group (PPG)?

Logical grouping that ensures VMs are physically close together in the same datacenter → reduces latency.

20. Real-world scenario: Your VM must handle unpredictable traffic spikes. What do you use?

Deploy VMs in a **VM Scale Set (VMSS)** with auto-scaling enabled.

Key Features

1. **Infrastructure as a Service (IaaS)** – Full control over OS, runtime, and applications.
2. **Wide OS Support** – Windows, Linux, SAP-certified OS, Oracle, Red Hat, Ubuntu, etc.
3. **VM Sizes (SKUs)** – General-purpose, compute-optimized, memory-optimized, storage-optimized, GPU, and HPC.
4. **Scalability** – Scale sets and availability sets provide elasticity and redundancy.
5. **Storage Flexibility** – Standard HDD, Standard SSD, Premium SSD, Ultra Disk options.
6. **Networking Integration** – VNet, NSG, Load Balancer, VPN Gateway, ExpressRoute.
7. **High Availability** – Availability Zones, Availability Sets, and fault domains.
8. **Security** – Microsoft Defender for Cloud, disk encryption, Secure Boot, TPM.
9. **Backup & DR** – Integrated with Azure Backup, ASR (Site Recovery).
10. **Automation** – Support for ARM, Bicep, Terraform, CLI, PowerShell.

Common Use Cases

1. **Lift-and-Shift Migration** – Move existing workloads to Azure without refactoring.
2. **Development & Testing** – Quick provisioning of test environments.
3. **Running Legacy Applications** – Host apps not yet compatible with PaaS.
4. **High-Performance Computing (HPC)** – Use GPU and compute-optimized VMs.
5. **Hosting Databases** – SQL Server, Oracle, MySQL on VMs.
6. **Disaster Recovery (DR)** – Backup and failover using Azure Site Recovery.
7. **Multi-Tier Applications** – Deploy front-end, application, and database tiers.
8. **Hybrid Scenarios** – Extend on-prem datacenters with Azure VMs.
9. **Custom Security Requirements** – Full control over OS hardening and patching.

10. **Virtual Desktop Infrastructure (VDI)** – Host desktops with Azure Virtual Desktop.

Best Practices

1. Use **availability zones/sets** for high availability.
2. Choose the **right VM size** for workload requirements (don't overprovision).
3. Use **Managed Disks** instead of unmanaged disks.
4. Enable **Azure Disk Encryption (ADE)** or Server-Side Encryption (SSE).
5. Always place VMs inside a **VNet with NSGs** for security.
6. Use **Managed Identities** instead of storing credentials in apps.
7. Apply **Auto-shutdown schedules** for dev/test VMs to save cost.
8. Monitor performance with **Azure Monitor & Log Analytics**.
9. Use **Azure Bastion or Just-in-Time VM access** (instead of exposing RDP/SSH publicly).
10. Regularly **patch & update OS** with Update Management.

3. Azure Fleet

1. What is Azure Fleet?

Azure **Fleet** is a service that allows you to manage, deploy, and scale **large heterogeneous groups of VMs** (different SKUs, Spot + Pay-as-you-go) for **large-scale compute workloads**. It's more flexible than **VM Scale Sets (VMSS)**, which are mostly for identical VM types.

👉 In short:

- **VMSS** → Scaling a single type of VM.
- **Fleet** → Scaling **multiple types** of VMs at once (multi-SKU, Spot + On-demand mix).

2. How is Azure Fleet different from VM Scale Sets?

- **VMSS** → Identical VM types, single SKU, app scaling.
- **Fleet** → Multiple SKUs, Spot + On-demand mix, large-scale compute.

3. When would you use Fleet over VMSS?

When workloads require **mixed VM types, Spot + On-demand blending, or massive-scale compute clusters** (e.g., HPC, AI training).

4. **Can Azure Fleet run across multiple regions?**
   Yes, it supports **multi-region deployments** to increase availability and reduce latency.

5. Does Azure Fleet guarantee VM allocation?
No, Spot capacity is not guaranteed. Fleet balances workloads across available SKUs and regions.

6. What VM allocation policies are available in Fleet?

- **Lowest Price** – Pick cheapest available Spot.
- **Capacity Optimized** – Choose SKUs with most availability.
- **Balanced** – Mix between price and availability.

7. What is the benefit of mixing Spot and On-demand VMs?
Spot reduces cost (up to 90%), On-demand ensures reliability. Fleet balances both.

8. How does Azure Fleet handle Spot eviction?
Fleet automatically **reallocates workloads** to available SKUs or On-demand VMs.

9. Can I use Reserved Instances with Azure Fleet?
Yes, Fleet can mix **RI, On-demand, and Spot** for cost efficiency.

10. How do you estimate cost in Azure Fleet?
By analyzing **VM SKU mix + Spot % + On-demand % + region pricing**.

11. How do you monitor Azure Fleet?
Using **Azure Monitor, Log Analytics, and Metrics** for allocation, eviction, and scaling.

12. How does scaling work in Azure Fleet?
Fleet supports **autoscaling policies** (metric or schedule-based).

13. Can Azure Fleet support GPU workloads?
Yes, it can mix CPU + GPU VM families.

14. How do you design applications for Spot workloads?

- Stateless architecture
- Checkpointing for ML training
- Retry + job queue mechanisms

15. What is a good workload example for Azure Fleet?

- AI model training with GPU + CPU mix
- Big data ETL pipelines
- Rendering pipelines

16. How does Azure Fleet improve resiliency compared to VMSS?
Fleet can failover to **different SKUs, families, and regions** if capacity is unavailable.

17. What are Proximity Placement Groups (PPGs) in context of Fleet?
They ensure VMs in a Fleet are **physically close** → useful for low-latency HPC workloads.

18. Can Azure Fleet integrate with Kubernetes (AKS)?
Yes, Fleet can provide **node pools** with mixed VM types for AKS clusters.

19. How do you secure VMs inside Azure Fleet?

- Use **VNet + NSGs**
- Enable **Azure Bastion/JIT access**
- Apply **disk encryption + managed identities**

20. Real-world scenario: You need 10,000 cores for a rendering job, but Spot VMs keep getting evicted. What's your solution?

- Use a **Spot + On-demand blend policy** (baseline On-demand, burst Spot).
- Distribute across **multiple SKUs/regions** for higher reliability.

Key Features of Azure Fleet

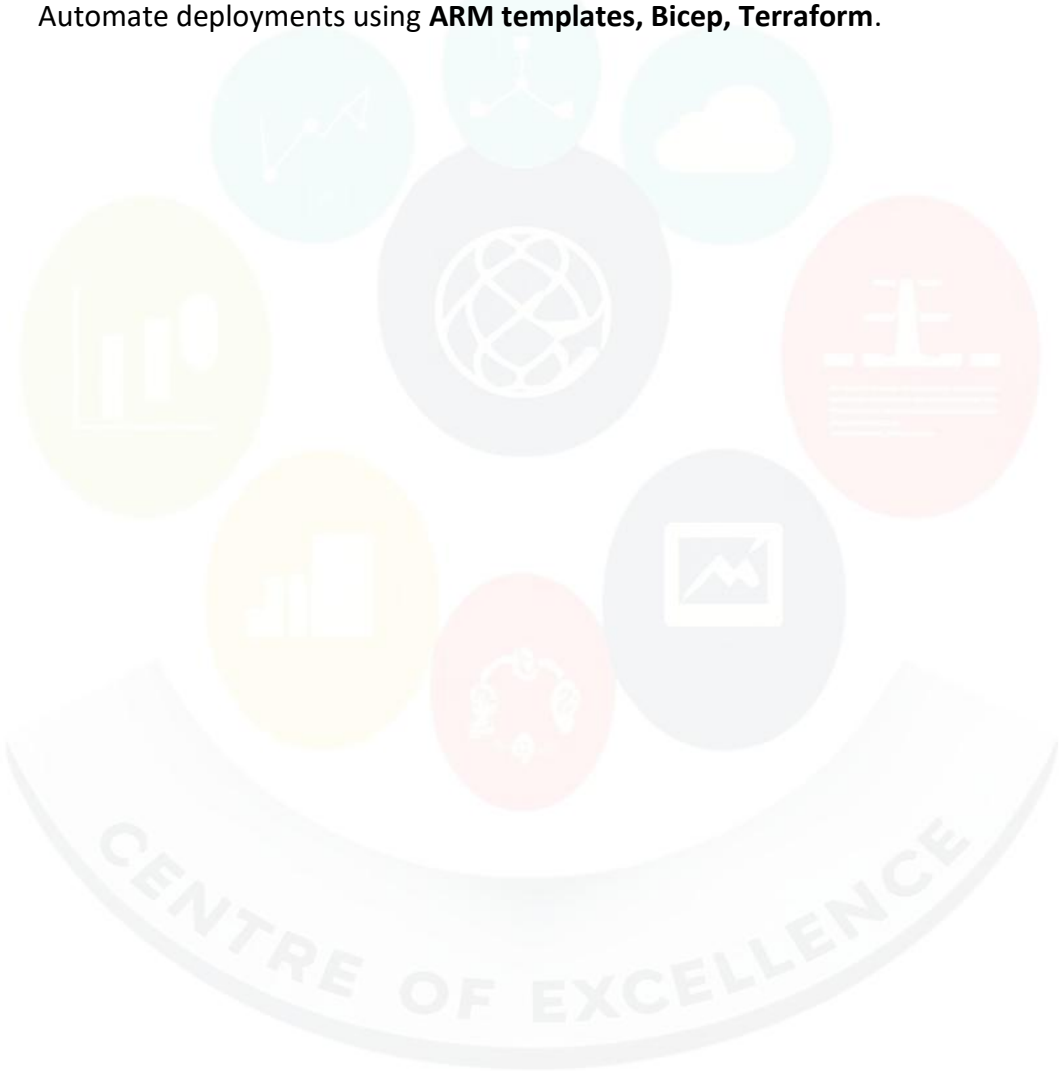
1. **Multi-SKU Support** – Mix different VM sizes, families, and SKUs in one fleet.
2. **Spot + On-Demand Mix** – Combine spot (cheap, preemptible) and pay-as-you-go VMs.
3. **Scale at Large** – Deploy **hundreds of thousands of VMs** across regions/zones.
4. **Custom Allocation Policies** – Define percentages of Spot vs On-Demand.
5. **Cross-Region Scaling** – Distribute workloads across multiple regions.
6. **Fault Tolerance** – Automatic rebalancing when Spot VMs are evicted.
7. **Workload Flexibility** – Suitable for batch jobs, HPC, AI/ML, rendering workloads.
8. **Cost Optimization** – Maximizes savings using Spot capacity first.
9. **Autoscaling** – Fleet expands or shrinks based on demand/metrics.
10. **API & IaC Support** – ARM templates, Bicep, Terraform, CLI support.

Common Use Cases

1. **HPC Workloads** – Scientific computing, weather simulations.
2. **AI/ML Training** – Large GPU + CPU mixed clusters.
3. **Big Data Processing** – ETL, Spark, Hadoop workloads.
4. **Rendering Jobs** – Video, 3D rendering workloads.
5. **Batch Jobs** – Parallel jobs requiring thousands of cores.
6. **Cost-Optimized Compute** – Mix Spot + On-demand to reduce cloud bills.
7. **Gaming Backends** – Dynamic game session scaling.
8. **Stress Testing & Load Testing** – Run short-lived compute-heavy workloads.
9. **Hybrid Workloads** – Mix memory-optimized + GPU VMs.
10. **Cloud Bursting** – Extend on-prem compute clusters to Azure Fleet.

Best Practices

1. Use **Spot VMs for non-critical workloads**, on-demand for baseline.
2. **Mix SKUs** for resilience → If one VM size is unavailable, others still provision.
3. Define **allocation strategies** → lowest price, capacity-optimized, or balanced.
4. Use **Autoscaling** to optimize performance & cost.
5. Deploy across **regions & zones** for fault tolerance.
6. Enable **Azure Monitor** for tracking evictions and fleet health.
7. Use **Proximity Placement Groups (PPG)** if low-latency clusters are needed.
8. Design workloads to handle **evictions gracefully** (stateless, checkpointing).
9. Use **managed identities** for secure access to resources.
10. Automate deployments using **ARM templates, Bicep, Terraform**.



5. Azure IP Address

What is an Azure IP Address?

An **IP address in Azure** is a unique numeric identifier assigned to Azure resources (like VMs, Load Balancers, Application Gateways) to enable communication within Azure (private) or with the internet (public).

1. What types of IP addresses are available in Azure?

- **Private IPs** → Internal communication inside VNets.
- **Public IPs** → External communication with internet.

2. What's the difference between Dynamic and Static IPs in Azure?

- **Dynamic** → Changes on VM stop/start.
- **Static** → Permanently assigned to resource.

3. What is the difference between Basic and Standard SKU Public IPs?

- **Basic** → Open by default, no zone redundancy.
- **Standard** → Secure by default (NSG required), zone-redundant, recommended for production.

4. Can Azure assign both IPv4 and IPv6 to a resource?

Yes, Azure supports **dual-stack IP configuration** (both IPv4 + IPv6).

5. How many private IP addresses can a VM NIC have?

Depends on **VM size**, but up to **250 IP configurations per NIC** are supported.

6. When should you use a Public IP vs a Private IP?

- **Private IP** → For internal-only apps.
- **Public IP** → When resource must be internet-facing.

7. What is a Public IP Prefix in Azure?

A **reserved contiguous range of public IPs** you can assign to multiple resources.

8. How do you secure a Public IP resource?

- Use **NSGs, Firewalls, WAFs**.
- Prefer **Standard SKU** with explicit rules.

9. What is NAT Gateway and how does it relate to Public IPs?

NAT Gateway provides **outbound internet connectivity** for VMs using **one or more static public IPs**, instead of assigning IPs to each VM.

10. What Azure resources require a Public IP?

- Load Balancer (Public)
- Application Gateway
- VPN Gateway
- Azure Firewall

- Azure Bastion (optional)

11. How does Azure assign Private IPs?
Via **DHCP within the VNet subnet range**. You can also assign static private IPs manually.

12. Can you move a Public IP from one resource to another?
Yes, A Public IP can be detached and reassigned to another resource.

13. What is IP forwarding in Azure NICs?
A setting that allows VMs (like firewalls, routers) to **forward traffic not addressed to them**.

14. How do you whitelist Azure services with IPs?
Use **Service Tags** (like Storage, Azure Cloud) instead of individual IPs.

15. What's the difference between an Instance-level Public IP (ILPIP) and a Load Balancer Public IP?

- **ILPIP** → Assigned directly to VM NIC.
- **LB Public IP** → Shared IP for multiple backend VMs via Load Balancer.

16. How do you ensure a web app hosted in a VM always uses the same outbound IP?

- Use **Static Public IP**.
- Or configure **NAT Gateway** with static IP.

17. How do Availability Zones impact IP addresses?
Standard SKU Public IPs can be **zonal or zone-redundant** for resiliency.

18. What happens to IP addresses when you stop and deallocate a VM?

- **Dynamic Public IPs** → Released.
- **Static Public IPs** → Retained.
- **Private IPs** → Retained unless VM is deleted.

19. Can an Azure VM have multiple NICs with different IPs?
Yes, Certain VM sizes support **multiple NICs**, each with multiple IPs.

20. Real-world scenario: You must allow only specific customers to access your API hosted on Azure. How do you configure IPs?

- Assign a **Static Public IP** to API.
- Configure **NSG/Firewall** to whitelist customer IP ranges.
- Optionally use **Azure Front Door** or **Application Gateway WAF**.

Features of Azure IP Address

1. **Private IPs** – Used for internal communication within VNets.

2. **Public IPs** – Used for internet-facing communication.
3. **Dynamic Allocation** – Assigned automatically and may change when a resource restarts.
4. **Static Allocation** – Permanently reserved to a resource, doesn't change on restart.
5. **IPv4 and IPv6 Support** – Both address versions supported.
6. **SKU Options** – Basic vs Standard (Standard has zone-redundancy, secure by default).
7. **Zonal/Zonal-Redundant** – High availability across availability zones.
8. **Public IP Prefixes** – Reserve a contiguous range of public IPs.
9. **DNS Labeling** – Public IPs can have DNS names for easier access.
10. **Integration with Services** – Required by Load Balancer, Application Gateway, VPN Gateway, Bastion, etc.

Common Use Cases

1. **Expose a Web App** → Assign a Public IP to VM or Load Balancer.
2. **Internal Communication** → Use Private IPs inside VNets.
3. **NAT Gateway** → Outbound traffic control via static public IPs.
4. **Application Gateway** → Needs Public IP for internet-facing apps.
5. **VPN Gateway** → Uses Public IP for site-to-site VPN.
6. **Multi-Region Deployments** → Public IP Prefixes for global reach.
7. **Secure Access** → Whitelisting via static IPs for databases/APIs.
8. **Hybrid Connectivity** → Public IP for ExpressRoute failover.
9. **Firewall Access** → Public IP for rules and whitelisting.
10. **API Endpoints** → Assign to APIs or services that must be reachable externally.

Best Practices

1. Use **Private IPs by default**, Public IPs only if required.
2. Prefer **Static IPs** for critical resources (DNS, gateways, load balancers).
3. Use **Standard SKU Public IPs** for production (secure by default, zone-resilient).
4. Use **NAT Gateway** instead of assigning individual Public IPs to VMs.
5. Avoid exposing VMs directly with Public IP → use Bastion or Load Balancer.
6. Leverage **NSGs + Firewall** to restrict access.
7. Use **IP Prefixes** for predictable IP ranges.
8. Plan **IP address ranges** in VNets carefully to avoid overlaps.
9. Use **IPv6** for apps requiring global reach or compliance.
10. Regularly audit Public IP usage to avoid unnecessary costs.

6. Azure Network Security Group

What is an NSG?

An **Azure Network Security Group (NSG)** is a security filter that controls **inbound and outbound traffic** to and from Azure resources (VMs, subnets, NICs) based on rules. Think of it as a **firewall at the network level**.

1. What is an NSG in Azure?

A **filtering mechanism** that allows or denies inbound and outbound traffic to Azure resources at **Layer 3 & 4** (IP, port, protocol).

2. What are the default rules in an NSG?

- Allow VNet inbound traffic.
- Allow Azure Load Balancer inbound.
- Deny all inbound from Internet.
- Allow VNet outbound traffic.
- Allow Internet outbound traffic.
- Deny all outbound traffic.

3. Where can NSGs be applied?

- At **Subnet level** (affects all resources in subnet).
- At **NIC level** (affects only that VM).

4. What is rule priority in NSGs?

Each rule has a **priority (100–4096)**. Lower number = higher precedence.

5. What is the difference between NSG and Azure Firewall?

- **NSG** → Basic network filtering (L3/L4).
- **Firewall** → Advanced filtering, L7 rules, TLS inspection, threat intelligence.

6. How do inbound and outbound rules work in NSGs?

- **Inbound rules** → Traffic entering VM/subnet.
- **Outbound rules** → Traffic leaving VM/subnet.

7. What is a Service Tag in NSGs?

A **predefined label** representing a group of IPs for Azure services (e.g., Internet, Storage, Azure Load Balancer).

8. What is an Application Security Group (ASG)?

A way to **group VMs logically** (e.g., “Web Servers”) to apply NSG rules without worrying about IP addresses.

9. Can NSG rules be stateful or stateless?
NSGs are **stateful** → if inbound is allowed, outbound response is automatically allowed.

10. How do you block all internet access for a subnet?
Create an outbound rule: **Deny Internet** on port * with lower priority than defaults.

11. How can you secure RDP/SSH using NSGs?

- Allow inbound RDP/SSH **only from whitelisted IPs**.
- Deny all other Internet access.

12. How do you log NSG traffic?
Enable **NSG Flow Logs** in **Network Watcher**, analyze via Log Analytics.

13. Can NSGs be used with Load Balancers?
Yes, NSGs can control traffic going **to/from LB frontend/backend pools**.

14. What happens if you apply an NSG at both NIC and Subnet?
Both rules apply. **Deny takes precedence** if conflicts exist.

15. How do you allow only a specific subnet to access a database VM?
Apply inbound NSG rule → Allow traffic from that subnet's IP range or ASG.

16. What's the maximum number of rules in an NSG?
Up to **1,000 rules per NSG** (inbound + outbound combined).

17. Can NSGs work across regions?
No, NSGs are **region-specific** → tied to VNets/subnets in a region.

18. How do you troubleshoot NSG issues?

- Use **Effective Security Rules** (portal).
- Use **Network Watcher IP Flow Verify**.
- Check **NSG Flow Logs**.

19. Can you combine NSG and Firewall?
Yes, Best practice:

- **NSG** for subnet-level filtering.
- **Firewall** for advanced, centralized security policies.

20. Real-world scenario: You host a 3-tier app (Web → App → DB). How do you design NSG rules?

- Web Subnet → Allow inbound HTTP/HTTPS from Internet.
- App Subnet → Allow only from Web subnet (ASG).
- DB Subnet → Allow only

Key Features of NSG

1. **Layer 3 & 4 Filtering** – Works on IP, port, and protocol.
2. **Inbound & Outbound Rules** – Define what traffic is allowed or denied.
3. **Applied to Subnets or NICs** – Flexible deployment.
4. **Default Rules** – Allow VNet traffic, Azure LB traffic, deny all inbound internet.
5. **Custom Rules** – Define based on IP, service tags, or application security groups.
6. **Rule Prioritization** – Lower priority = higher precedence.
7. **Service Tags** – Predefined tags (e.g., Internet, VirtualNetwork, AzureLoadBalancer).
8. **Application Security Groups (ASGs)** – Group VMs logically instead of by IP.
9. **Stateless Behavior** – If inbound is allowed, outbound response is automatically allowed.
10. **Logging & Monitoring** – NSG flow logs via Azure Monitor & Log Analytics.

Common Use Cases

1. **Restrict RDP/SSH Access** → Allow only from specific IPs.
2. **Secure VM Traffic** → Control inbound/outbound VM access.
3. **Subnet-Level Protection** → Apply NSG to isolate workloads.
4. **Web App Security** → Allow HTTP/HTTPS, deny everything else.
5. **Database Security** → Allow only app subnet to connect to DB subnet.
6. **Hybrid Connectivity** → Whitelist on-prem IPs for VPN/ExpressRoute traffic.
7. **Multi-Tier App Architecture** → Apply rules at each tier (Web → App → DB).
8. **Restrict Outbound Internet Access** → Prevent VMs from reaching external IPs.
9. **Micro-Segmentation** → Use ASGs to control traffic between groups of VMs.
10. **Compliance & Auditing** → Log traffic for audits.

Best Practices

1. Apply NSGs at **subnet level**, NIC NSGs only for exceptions.
2. Use **ASGs** instead of IP addresses for dynamic grouping.
3. Use **Service Tags** instead of hardcoding Azure IP ranges.
4. Deny **RDP/SSH from Internet**, use Bastion or JIT access.
5. **Log and monitor** traffic with NSG flow logs in Log Analytics.
6. Keep rules minimal → deny by default, allow explicitly.
7. Use **priority planning** (lower number = higher precedence).
8. Separate inbound and outbound control clearly.
9. Combine with **Azure Firewall** for advanced filtering.
10. Regularly review NSG rules to remove unused entries.

7. Azure NIC

What is an Azure NIC?

An **Azure Network Interface Card (NIC)** is a networking component that connects a **Virtual Machine (VM)** to a **Virtual Network (VNet)**. Every VM needs at least one NIC to communicate within Azure or with the internet.

1. What is an Azure NIC?
A NIC connects an Azure VM to a **VNet** and provides it with IP addressing and network security.

2. Can a VM have multiple NICs?
Yes, VM sizes determine NIC limits. Some VMs support up to **8 NICs**.

3. What is the difference between a primary and secondary NIC?

- **Primary NIC** → Required, default network interface.
- **Secondary NICs** → Optional, used for additional traffic separation.

4. Can you detach and reattach a NIC to a VM?
Yes, but only **secondary NICs** can be detached while the VM is running. The **primary NIC** cannot be removed.

5. What is the role of MAC addresses in NICs?
Each NIC has a **unique MAC address** for Ethernet-level identification.

6. How many IP addresses can a NIC have?

- One **primary private IP**.
- Multiple **secondary private IPs** (up to 250 per NIC depending on VM size).
- Each private IP can optionally have a **public IP**.

7. Difference between Static and Dynamic private IPs on NICs?

- **Dynamic** → Assigned automatically, can change if VM is stopped/deallocated.
- **Static** → Permanently assigned, doesn't change on restart.

8. Can a NIC have multiple public IPs?
Yes, via multiple IP configurations.

9. What is Accelerated Networking in NICs?
Feature that uses **SR-IOV** to bypass host network stack → lowers latency, improves throughput.

10. Can I assign IPv6 to an Azure NIC?
Yes, NICs support **dual-stack IPs (IPv4 + IPv6)**.

11. How do NSGs work with NICs?
NSGs can be applied at **subnet or NIC level**. Both are evaluated, and **deny rules take precedence**.

12. What happens if a subnet NSG and NIC NSG have conflicting rules?
The **most restrictive rule (deny)** is enforced.

13. Can I move a NIC from one VNet to another?
No, A NIC is bound to a **single VNet**. You must recreate it in the new VNet.

14. How do you monitor NIC traffic in Azure?

- **NSG Flow Logs**
- **Network Watcher Packet Capture**
- **Metrics in Azure Monitor**

15. Can I assign multiple NICs to different subnets?
Yes, multi-NIC VMs can connect to different subnets in the same VNet.

16. How does Azure NIC help in creating DMZs?
A VM can have one NIC in a **public-facing subnet** and another NIC in a **private subnet**.

17. Why would you use multiple NICs on a VM?

- Traffic separation (frontend/backend).
- Network Virtual Appliances (firewalls, routers).
- Different subnets for management and application traffic.

18. What is IP forwarding in NICs?
A NIC setting that allows a VM to act as a **router/firewall**, forwarding packets not destined to it.

19. Can NICs be used with Load Balancers?
Yes, NICs in a backend pool of a Load Balancer receive traffic via load balancing rules.

20. Real-world scenario: You host a firewall appliance VM in Azure. How do you configure its NICs?

- **One NIC** in a **public subnet** (for internet traffic).
- **One NIC** in a **private subnet** (for internal traffic).
- Enable **IP forwarding** to allow traffic routing.

Key Features of Azure NIC

1. **VM Attachment** – A NIC must be attached to a VM for it to communicate.
2. **Primary & Secondary NICs** – A VM can have one **primary NIC** and multiple **secondary NICs** (depending on VM size).

3. **Multiple IP Configurations** – A NIC can have **multiple IP addresses** (private + public).
4. **Private IP Assignment** – Supports **static or dynamic** private IPs.
5. **Public IP Assignment** – Can have public IPs associated via IP configuration.
6. **NSG Association** – Security filtering at NIC level.
7. **Accelerated Networking** – Improves throughput and reduces latency (SR-IOV).
8. **MAC Address** – Each NIC gets a unique MAC address.
9. **Diagnostics** – Monitor with Network Watcher.
10. **Flexibility** – A NIC can be detached from one VM and re-attached to another.

Common Use Cases

1. **VM Communication** – Provides networking capability for VMs.
2. **Multi-NIC Architecture** – Use separate NICs for frontend and backend traffic.
3. **Load Balancing** – Attach Public IP to NIC for internet-facing workloads.
4. **High Security Apps** – Apply NSGs directly at NIC level.
5. **DMZ Scenarios** – One NIC connected to internet-facing subnet, another to private subnet.
6. **Multiple IPs** – Run multiple apps/services on a single VM.
7. **Accelerated Networking** – Improve performance for latency-sensitive apps.
8. **Network Virtual Appliances (NVA)** – Firewalls/routers often use multiple NICs.
9. **Failover & Resiliency** – Reassign NICs between VMs in DR scenarios.
10. **Traffic Isolation** – Separate management, app, and storage traffic.

Best Practices

1. Use **subnet-level NSGs**, NIC-level NSGs only for exceptions.
2. Plan **IP allocation** (avoid overlapping private IPs).
3. Use **static private IPs** for critical VMs (databases, domain controllers).
4. Enable **Accelerated Networking** for production workloads.
5. Minimize **public IPs on NICs** → use Bastion, Load Balancer, or NAT Gateway.
6. Use **multi-NIC** only if workload requires traffic separation.
7. Monitor NIC performance with **Network Watcher**.
8. Avoid unnecessary NIC detachment/reattachment (can cause downtime).
9. Use **Application Security Groups (ASGs)** instead of IP-based rules.
10. Tag NICs for cost tracking and management.

8. Azure Disk Storage

Azure Disks are **block-level storage volumes** used as persistent storage for Azure Virtual Machines (VMs). They store the **OS**, **application data**, and **user data**.
Azure Disks are highly available, durable, and managed by Azure (Managed Disks).

1. What are Azure Disks?
Azure Disks are block-level storage volumes attached to VMs for OS, apps, and data.

2. What's the difference between Managed and Unmanaged Disks?

- **Managed Disks** → Azure manages storage accounts, scaling, reliability.
- **Unmanaged Disks** → User manages storage accounts (legacy, not recommended).

3. What are the different types of Azure Disks?

- **Ultra Disk** → Ultra-low latency, high IOPS workloads.
- **Premium SSD** → Mission-critical production apps.
- **Standard SSD** → Reliable mid-tier performance.
- **Standard HDD** → Cost-effective, dev/test workloads.

4. What is the max size of Azure Disks?

- **OS Disk** → Up to 4 TB.
- **Data Disk** → Up to TB.

5. What is a Temporary Disk in Azure?
Each VM has a temporary disk (D: in Windows, /dev/sdb in Linux) for page/cache data. **Not durable** → data lost on VM restart/deallocation.

6. How do you increase IOPS for Azure Disks?

- Use **Premium SSD or Ultra Disk**.
- Attach multiple disks and use **disk striping**.
- Use **disk bursting** for short-term spikes.

7. What's the difference between caching modes (None, ReadOnly, ReadWrite)?

- **None** → No caching.
- **ReadOnly** → Speeds up read-heavy workloads (DB, web servers).
- **ReadWrite** → Useful for OS disk boot performance.

8. Which disk type supports bursting?

- Premium SSD (P20 or smaller).
- Standard SSD (E30 or smaller).

9. What is Ultra Disk best suited for?
Databases requiring sub-ms latency and high throughput (SAP HANA, SQL Server, Oracle).

10. How do you handle large IOPS requirements?

- Use **multiple data disks** + **RAID 0 striping**.
- Choose **VMs that support high IOPS** (e.g., M-series, Lsv2).

11. How are Azure Disks replicated?

- **LRS (Locally Redundant Storage)** – 3 replicas in one datacenter.
- **ZRS (Zone-Redundant Storage)** – Replicated across Availability Zones.

12. How do you encrypt Azure Disks?

- **Azure Disk Encryption (ADE)** → Encrypts with BitLocker (Windows) / DM-Crypt (Linux).
- **Server-Side Encryption (SSE)** → Encrypts data at rest with Azure-managed keys or customer-managed keys in Key Vault.

13. Can you take snapshots of disks?

Yes, Snapshots allow point-in-time backup and restore of disks.

14. What is the difference between snapshot and image?

- **Snapshot** → Copy of a single disk.
- **Image** → Full VM image (OS + data disks).

15. Can Azure Disks be shared between VMs?

Yes, **Shared Disks** feature allows multiple VMs to attach to the same disk (for clustered apps like SQL).

16. How do you migrate an on-prem VHD to Azure?

Upload the VHD to Azure Blob Storage → Create Managed Disk → Attach to VM.

17. Can you resize an Azure Disk?

Yes, Disks can be resized **upwards** (not downwards).

18. How do you back up Azure Disks?

Using **Azure Backup** service or **manual snapshots**.

19. Real-world scenario: Which disk type should you use for a production SQL Database VM?

- **Premium SSD** (general).
- **Ultra Disk** (extreme performance needs).

20. What's the difference between Standard SSD and Premium SSD?

- **Standard SSD** → Cheaper, reliable, higher latency.
- **Premium SSD** → Higher IOPS, low latency (<1ms), better for production.

Key Features of Azure Disks

1. **Types of Disks** – Ultra Disk, Premium SSD, Standard SSD, Standard HDD.
2. **OS Disk & Data Disk** –
    - OS Disk → Boot volume (up to 4 TB).
    - Data Disks → Attached for extra storage (up to 32 TB each).
3. **Temporary Disk** – Local VM disk (D: drive in Windows) for page/cache data (not durable).
4. **Managed Disks** – Azure manages storage accounts, replicas, scaling.
5. **Unmanaged Disks (legacy)** – User-managed storage accounts.
6. **Snapshots & Images** – Point-in-time copies of disks.
7. **Encryption** – Azure Disk Encryption (ADE) & Server-Side Encryption (SSE).
8. **Availability** – 3x replicas within the region; ZRS for zone-redundant storage.
9. **Performance Tiers** – Different IOPS & throughput based on disk type.
10. **Scalability** – Up to 50,000 IOPS per VM with multiple disks.

Common Use Cases

1. **OS Boot Volume** – Every VM needs an OS Disk.
2. **Database Storage** – Premium SSD/Ultra Disk for SQL Server, Oracle, SAP HANA.
3. **Big Data & Analytics** – High IOPS disks for Spark, Hadoop workloads.
4. **File Storage & Applications** – Hosting application binaries, logs.
5. **Dev/Test Workloads** – Standard HDD or Standard SSD for low-cost testing.
6. **Disaster Recovery** – Snapshots for backup and restore.
7. **Lift-and-Shift Migrations** – Import/export VHDs to Azure.
8. **Caching & Temp Storage** – Using temporary disk for swap/cache.
9. **Multiple Disks Striping** – Increase throughput by attaching multiple disks.
10. **High Availability Apps** – Zone-redundant disks across Availability Zones.

Best Practices

1. Use **Premium SSD/Ultra Disk** for production workloads requiring low latency.
2. Use **Standard SSD/HDD** for dev/test and backup scenarios.
3. **Enable disk caching** (ReadOnly/ReadWrite) to boost performance for suitable workloads.
4. Use **Azure Backup/Snapshots** for point-in-time recovery.
5. Encrypt disks using **Azure Key Vault + ADE/SSE**.
6. Use **ZRS Disks** for cross-zone availability.
7. Use **disk bursting** feature for temporary performance spikes.
8. Choose **Ultra Disk** for sub-millisecond latency workloads.
9. **Separate OS and Data disks** – avoid running apps on OS disk.
10. Use **disk striping** (RAID 0 at VM level) for higher throughput when needed.

9. Azure Bastion

Azure Bastion is a **fully managed PaaS service** that allows secure and seamless **RDP/SSH access to Azure Virtual Machines** directly from the Azure portal over SSL, **without exposing public IP addresses**.

It eliminates the need for a **jump server** or VPN just for remote access.

1. What is Azure Bastion?
Azure Bastion is a PaaS service that provides secure RDP/SSH connectivity to Azure VMs **without exposing public IPs**, via the Azure portal.

2. How is Bastion different from a traditional jump server?

- Bastion is **managed by Azure**, scales automatically, and requires **no maintenance**.
- Jump servers need patching, monitoring, and are attack-prone.

3. Which protocol does Bastion use for connectivity?
Bastion connects over **SSL (port 443)**, making it firewall-friendly.

4. Does Bastion require a public IP on the VM?
No, VMs can remain private within the VNet.

5. Can Bastion connect to VMs in different VNets?
Yes, If VNets are **peered** or if using a **hub-spoke network**.

6. What are the two SKUs of Bastion?

- **Basic** → For small-scale deployments.
- **Standard** → Supports auto-scaling, native client, Kerberos, IP-based connections.

7. How is authentication managed in Bastion?
Via **Azure RBAC** roles and optionally **Azure AD authentication** (Standard).

8. Can multiple users connect to the same VM using Bastion?
Yes, Bastion supports concurrent sessions.

9. What are some advanced features available in Bastion Standard SKU?

- Azure AD authentication.
- Native client (mstsc/ssh).
- Kerberos authentication.
- IP-based connections.

10. How does Bastion improve security compared to opening RDP/SSH ports?
It prevents brute-force and malware attacks on **public IPs**, since ports 22/3389 are never exposed.

11. How is Bastion deployed?
In a **dedicated subnet** called AzureBastionSubnet within the VNet.

12. Can you use Bastion with NSGs and firewalls?
Yes, Bastion traffic must be allowed on port 443 to Azure service tags.

13. Can you restrict Bastion usage with policies?
Yes, via **Azure Policy** and **RBAC roles**.

14. How do you monitor Bastion activity?
Using **Azure Monitor, Activity Logs, and Network Watcher**.

15. Is Bastion HA by default?
Yes, Bastion is a PaaS service with built-in high availability.

16. Can you use Bastion without the Azure Portal?
Yes, Standard SKU supports **native RDP/SSH clients**.

17. Can Bastion connect to on-prem VMs?
Not directly, Only to Azure VMs inside a VNet. For on-prem, you'd need VPN/ExpressRoute.

18. What happens if you delete a Bastion Host?
VMs remain unaffected, but you lose the Bastion-based access method.

19. Real-world scenario: Your security team wants to block RDP port 3389 and SSH port 22 from the internet. How can you still manage VMs?
Deploy **Azure Bastion** → access VMs securely via portal.

20. How does Bastion integrate into a hub-spoke architecture?
Deploy Bastion in **hub VNet**, peer spoke VNets, and manage all VMs centrally.

Key Features of Azure Bastion

1. **Browser-based RDP/SSH** – Access VMs directly from the Azure Portal.
2. **No Public IP Required** – VMs remain private within the VNet.
3. **TLS/SSL Connectivity** – Secured via port 443.
4. **Integration with Azure RBAC** – Role-based access control.
5. **Multi-session Support** – Multiple users can connect simultaneously.
6. **Clipboard Functionality** – Copy-paste between local machine and VM session.
7. **Full Agentless Access** – No software or agent needed on VM.
8. **Scale-out Support** – Automatically scales with user load.
9. **Advanced SKU** – Offers features like IP-based connection, Kerberos authentication, native client support.
10. **High Availability** – Built-in redundancy.

Common Use Cases

1. **Secure VM Management** – Access VMs without exposing RDP/SSH ports.
2. **Compliance** – Restrict inbound internet traffic while allowing management access.

3. **Remote Workforce** – Admins can manage VMs securely from anywhere.
4. **Jump Server Replacement** – Removes need for Bastion Hosts/Jump boxes.
5. **Multi-user Admin Access** – Multiple admins can connect simultaneously.
6. **Hybrid Cloud Security** – Works well in enterprises with strict firewall policies.
7. **Temporary Access** – Grant on-demand access for contractors.
8. **Advanced Bastion** – Supports Azure AD authentication for compliance-heavy orgs.

Best Practices

1. Always use **Azure Bastion** instead of exposing public RDP/SSH ports.
2. Deploy Bastion in the **same VNet as the VMs** to manage.
3. Use **Network Security Groups (NSGs)** to restrict Bastion subnet access.
4. Use **Azure AD-based authentication** with Bastion (advanced tier).
5. For large-scale environments, enable **autoscale in Bastion Standard SKU**.
6. Restrict **clipboard sharing** if security requires.
7. Place Bastion in a **hub VNet** for shared access across spoke VNets.
8. Monitor Bastion usage with **Azure Monitor** and **Activity Logs**.
9. Combine with **Just-in-Time (JIT) VM access** for additional security.
10. Use **Private Bastion** if traffic should remain fully internal.

10. Azure Virtual Machine Scale Set

1. **What is Azure Virtual Machine Scale Set (VMSS)?**
   VMSS is a service that allows you to deploy and manage a set of identical (or mixed) VMs that automatically scale in or out based on demand.

2. **How is VMSS different from an Availability Set?**

   - **Availability Set:** Provides HA by spreading VMs across FDs and UDs, but scaling is manual.
   - **VMSS:** Provides **autoscaling + HA**, with uniform or flexible orchestration.

3. **What are orchestration modes in VMSS?**
   - **Uniform Orchestration:** All VMs identical, managed as a group.
   - **Flexible Orchestration:** Supports multiple VM sizes/SKUs, ideal for microservices and mixed workloads.

4. **How does VMSS provide high availability?**
   VMs are distributed across **fault domains, update domains, and optionally availability zones** to minimize downtime during failures or maintenance.

5. **How does autoscaling work in VMSS?**
   VMSS uses **Azure Autoscale** rules:
   - Metric-based (CPU, memory, disk, custom)
   - Schedule-based (e.g., scale up at 9 AM, down at 6 PM)

6. **Can VMSS use custom VM images?**
   Yes, you can use **custom images** from **Shared Image Gallery** or marketplace images.

7. **How does rolling upgrade work in VMSS?**
   - Updates are applied to VMs in batches (controlled %).
   - Ensures some VMs remain available while others are upgraded.

8. **Can you use Spot VMs with VMSS?**
   Yes, in **flexible orchestration mode**, you can mix Spot + pay-as-you-go VMs for cost savings.
9. **How do you distribute traffic across VMSS instances?**
   By integrating with **Azure Load Balancer** (Layer 4) or **Application Gateway** (Layer 7).

10. **Can VMSS integrate with Availability Zones?**
    Yes, VMSS can span across zones for **datacenter-level resiliency**.

11. **Difference between VMSS and Kubernetes (AKS)?**
    - **VMSS:** Focus on scaling VMs, infrastructure-level.
    - **AKS:** Container orchestration, application-level scaling.
      👉 Often AKS worker nodes run **inside a VMSS**.

12. **Difference between VMSS and Autoscale for App Service?**

- **VMSS:** For VM-based workloads (IaaS).
- **App Service Autoscale:** For PaaS web apps.

13. **When should you use Flexible vs Uniform Orchestration?**
- **Uniform:** Best for identical workloads (e.g., web servers).
- **Flexible:** Best for mixed workloads (e.g., microservices, spot + regular VM mix).

14. **Can VMSS handle stateful workloads?**
Yes, VMSS supports **data disks** and **stateful VMs** with **automatic instance repair**.

15. **How does VMSS handle failed VMs?**
- Azure automatically **reimages or replaces unhealthy instances** based on health probes.

16. **How do you secure VMSS instances?**

- Use **Managed Identity** for access to other resources.
- Restrict RDP/SSH with **Just-In-Time access**.
- Apply **NSGs and Azure Firewall**.

17. **What is the maximum number of VMs in a VMSS?**
- Up to **1,000 instances per scale set** (with managed disks).

18. **How can you monitor a VMSS?**
- **Azure Monitor metrics** (CPU, memory, disk).
- **Log Analytics** for deep analysis.
- Alerts on scaling events and failures.

19. **What's the role of Shared Image Gallery with VMSS?**

- Ensures **consistent golden images** across thousands of VMSS instances.
- Supports **regional replication** for global workloads.

20. Best strategy: When to use VMSS vs Availability Zones?
- **VMSS:** For autoscaling and elasticity.
- **Availability Zones:** For regional resiliency.
  👉 Best practice: **Use VMSS across Availability Zones** for both **scalability + resiliency**.

Key Features

1. **Automatic Scaling** – Scale in/out based on demand (CPU, memory, custom metrics).
2. **Uniform or Flexible Orchestration** – Identical VMs (uniform) or multiple SKUs (flexible).
3. **Integration with Load Balancer & App Gateway** – Distributes traffic across VM instances.
4. **High Availability & Fault Tolerance** – VMs spread across **Fault Domains** and **Update Domains**.
5. **Supports Custom Images & Marketplace Images** – Use your own golden image or gallery.
6. **Rolling Upgrades** – Apply updates without downtime.
7. **Autoscaling Rules** – Schedule or metric-based scaling.
8. **Integration with Availability Zones** – Place VMs across zones for better resiliency.
9. **Support for Spot VMs** – Cost optimization for non-critical workloads.
10. **Azure Monitor Integration** – Metrics, autoscaling, and alerting built-in.

Common Use Cases

1. **Web Applications** – Scale web servers during traffic spikes.
2. **API Backends** – Handle fluctuating load automatically.
3. **Big Data Processing** – Spin up large compute clusters on demand.
4. **High-Performance Computing (HPC)** – Scientific workloads requiring massive compute.
5. **Dev/Test Environments** – Scale environments up or down as needed.
6. **E-commerce Applications** – Handle seasonal or flash-sale demand.
7. **Gaming Servers** – Handle player surges dynamically.

Best Practices

1. Use **autoscaling rules** based on **custom metrics (e.g., queue length)** instead of only CPU.
2. Spread across **Availability Zones** for better resiliency.
3. Use **Azure Managed Disks** for improved performance and reliability.
4. Always integrate with a **Load Balancer/App Gateway**.
5. Optimize **instance size and SKU mix** for cost-performance balance.
6. Use **custom images with Shared Image Gallery** for consistency.
7. Enable **health probes & automatic instance repair**.
8. Use **Spot VMs in flexible orchestration** for cost-sensitive workloads.
9. Monitor with **Azure Monitor + Log Analytics**.
10. Automate with **ARM templates, Terraform, or Bicep**.

11. Azure Availability Set

2. **What is an Availability Set in Azure?**
   An Availability Set is a logical grouping of VMs that ensures Azure places them across different **fault domains (hardware racks)** and **update domains (maintenance groups)** to provide **high availability**.

3. **Why do we need an Availability Set?**
   To protect applications from **hardware failures** and **planned maintenance downtime**, ensuring at least one VM remains available.

4. **What SLA does Azure provide with Availability Sets?**
   99.95% uptime SLA when 2 or more VMs are placed in an availability set.

5. **What is the difference between Fault Domain and Update Domain?**
   - **Fault Domain (FD):** Physical separation (server racks, power, network).
   - **Update Domain (UD):** Logical group; Azure updates one UD at a time during patching.

6. **How many FDs and UDs can an availability set have?**
   - Fault Domains: Up to **3** (default).
   - Update Domains: Up to **20** (configurable).

6. Can you add a VM to an Availability Set after creation?
No. Once created, a VM cannot be moved into an availability set. You must recreate the VM.

7. How does Azure ensure VM distribution in an availability set?
Azure automatically spreads VMs across **FDs and UDs** to minimize downtime.

8. What happens if you deploy only one VM in an availability set?
You do **not** get any high availability benefits — SLA is still only **99.9%**, same as a single VM.

9. Can Availability Sets span across multiple regions?
No. They are limited to a **single datacenter region**. For cross-region HA, use **Availability Zones** or **Geo-Replication**.

10. Can Availability Sets be used with Virtual Machine Scale Sets (VMSS)?
No. VMSS already provide built-in distribution across FDs and UDs.

11. Difference between Availability Set and Availability Zone?

Feature	Availability Set	Availability Zone

Scope	Single datacenter	Multiple datacenters in a region
Protection	Rack/power/network failure	Datacenter failure
SLA	99.95%	99.99%
Cost	Free	Slightly higher (zone costs)

12. Difference between Availability Set and Load Balancer?

- **Availability Set:** Ensures VM uptime during failures/updates.
- **Load Balancer:** Distributes traffic between available VMs.

👉 Best practice: Use **both together**.

13. How does an Availability Set handle OS updates?

Azure updates **one UD at a time** so not all VMs reboot simultaneously.

14. In what scenario would you use Availability Set instead of Availability Zone?

- When **Availability Zones are not supported** in your region.
- When protection from **rack-level failure** is enough for the workload.

15. Can you use Availability Set for SQL Server Always On?

Yes. Place database nodes in an availability set to ensure HA inside a region.

16. What are the cost implications of Availability Sets?

- Availability Sets are **free**, but you pay for the VMs and supporting resources.

17. How do Managed Disks improve Availability Sets?

Managed disks are placed in **different storage scale units**, ensuring no single storage fault affects all VMs.

18. How can you monitor Availability Set health?

- Use **Azure Monitor & Metrics** (e.g., VM uptime).
- Set alerts on **VM failures** and **platform events**.

19. What is the maximum number of VMs in an Availability Set?

- Azure allows up to **200 VMs** per availability set.

20. Best strategy: Should I choose Availability Sets or Availability Zones?

- **Availability Sets:** Good for **intra-datacenter HA** (basic workloads).
- **Availability Zones:** Better for **mission-critical, production-grade apps** needing datacenter-level resiliency.

Key Features

1. **High Availability (HA)** – Protects VMs against single points of failure in Azure datacenter.
2. **Fault Domains (FDs)** – Physical separation of hardware (power, network, racks).
3. **Update Domains (UDs)** – Logical groups to ensure updates don't reboot all VMs at once.
4. **99.95% SLA** – Guaranteed uptime when 2+ VMs are deployed in an availability set.
5. **Automatic Distribution** – Azure spreads VMs across FDs and UDs.
6. **Works only for VMs** – Not applicable to PaaS services.
7. **Load Balancer Integration** – Commonly used with Azure Load Balancer for resiliency.

Common Use Cases

1. **Web Servers** – Multiple VMs in a set ensure one is always available.
2. **Application Servers** – Redundant instances to handle patching and failures.
3. **Database Clusters** – SQL Always On nodes distributed across domains.
4. **On-Premises Failover Replacement** – Cloud-native high availability.
5. **Enterprise Production Apps** – Ensures compliance with HA SLAs.

Best Practices

1. Always deploy **at least 2 VMs** in an availability set.
2. Combine with **Load Balancer** for traffic distribution.
3. Use **Managed Disks** for improved fault isolation.
4. Plan **number of UDs & FDs** based on workload needs.
5. Monitor VM placement using **Azure Resource Explorer**.
6. Use **Availability Zones** (if possible) for even higher resiliency.
7. Don't mix **prod & non-prod VMs** in the same availability set.
8. Use **automation (ARM templates/Terraform)** for consistent setup.

12. Azure Availability Zone

1. What are Azure Availability Zones?
Physically separate datacenters within an Azure region, each with independent power, cooling, and networking, designed to protect against **datacenter-level failures**.

2. How many Availability Zones exist in each Azure region?
Typically **3 zones per supported region** (Zone 1, Zone 2, Zone 3).

3. What SLA does Azure provide with Availability Zones?

- **99.99% SLA** when 2+ VMs are deployed across multiple zones with a Load Balancer.

4. How do Availability Zones differ from Availability Sets?

Feature	Availability Set	Availability Zone
Scope	Rack-level protection	Datacenter-level protection
SLA	99.95%	99.99%
Coverage	Single datacenter	Multiple datacenters in a region

5. How does Azure ensure communication between zones?
Azure provides **high-bandwidth, low-latency private network connectivity** between zones in the same region.

6. Can you deploy a VM in a specific Availability Zone?
Yes. When creating a VM, you can specify the **zone number (1, 2, or 3)**.

7. What is Zonal vs Zone-Redundant resources?

- **Zonal:** Resource is pinned to a specific zone (e.g., VM, IP).
- **Zone-Redundant (ZRS):** Replicates across zones (e.g., Azure Storage ZRS, SQL Database).

8. Can all Azure regions support Availability Zones?
No. Only **selected regions** have Availability Zones.

9. What happens if one zone goes down?

- Zonal resources in that zone go down.
- Zone-redundant resources (ZRS) continue functioning.
- Load Balancer routes traffic to healthy zones.

10. How does pricing work for Availability Zones?

- No extra cost for zones themselves.
- You pay for deployed resources and **inter-zone data transfer** (if applicable).

11. Difference between Availability Zones and Regions?

- **Region:** Geographical area (e.g., East US).
- **Availability Zone:** Multiple datacenters **within a region**.

12. Difference between Availability Zone and Disaster Recovery (DR)?

- **Availability Zone:** Protects against local datacenter outages.
- **DR (Geo-Replication):** Protects against **regional failures**.

13. When would you use Availability Zone vs Availability Set?

- **Availability Set:** For rack-level protection, less critical apps.
- **Availability Zone:** For mission-critical apps requiring datacenter-level HA.

14. How do Availability Zones integrate with Load Balancer?

Azure Load Balancer distributes traffic to VMs in multiple zones, ensuring zone failure doesn't disrupt service.

15. Can you deploy SQL Always On in Availability Zones?

Yes. Each SQL node can be in a different zone for high availability.

16. How do you monitor Availability Zone deployments?

- **Azure Monitor** → VM health, zonal metrics.
- **Log Analytics** → zone failure analysis.

17. What is ZRS in Azure Storage?

Zone-Redundant Storage (ZRS): Replicates data across 3 zones in a region, ensuring durability even if a zone fails.

18. How do Availability Zones impact latency?

- Inter-zone latency is **low but non-zero**.
- Design apps to handle **cross-zone network calls** efficiently.

19. Can Availability Zones be combined with Availability Sets?

Yes. A VM inside a **zone** can also be part of an **availability set** for extra redundancy.

20. Best strategy: Availability Zones vs Region Pairs?

- **Availability Zones:** For **high availability within a region**.

- **Region Pairs:** For **disaster recovery across regions**.
  👉 Best practice: Use **both together** for mission-critical apps.

Key Features

1. **Physically Separate Data Centers** – Each zone has independent power, cooling, and networking.
2. **High Availability** – Protects against entire datacenter failure.
3. **99.99% SLA** – When VMs are deployed across zones with Load Balancer.
4. **Zonal Services** – Resources pinned to a specific zone (VMs, disks, IPs).
5. **Zone-Redundant Services (ZRS)** – Replicate data across zones (e.g., Azure Storage ZRS, SQL DB, Event Hub).
6. **Integration with Load Balancer & Traffic Manager** – Automatic failover across zones.
7. **Supports Multi-Region Resiliency** – Combine with paired regions for disaster recovery.
8. **Low-Latency Networking** – High-bandwidth inter-zone connectivity.
9. **No Extra Cost for Zones** – You pay only for deployed resources.
10. **Improves Business Continuity** – Meets compliance for mission-critical apps.

Common Use Cases

1. **Mission-Critical Applications** – Banking, healthcare, ERP systems.
2. **E-commerce Platforms** – Ensure uptime during heavy seasonal traffic.
3. **Databases** – Deploy SQL Always On across zones for HA.
4. **Enterprise SaaS Solutions** – Guarantee uptime for customers.
5. **Disaster Recovery Strategy** – Zone-to-zone failover in the same region.
6. **Kubernetes (AKS)** – Deploy worker nodes across multiple zones.
7. **Financial Transactions** – Achieve resiliency requirements for compliance.

Best Practices

1. Always deploy **at least 2 or 3 VMs** across zones.
2. Combine with **Azure Load Balancer / Application Gateway** for traffic distribution.
3. Use **ZRS (Zone-Redundant Storage)** for critical data.
4. Design apps to be **zone-aware** (stateless where possible).
5. Automate deployments using **ARM templates, Terraform, Bicep**.
6. Monitor zonal distribution using **Azure Resource Explorer**.
7. For ultra-critical workloads, combine **Availability Zones + Paired Regions**.
8. Don’t assume **latency is zero** between zones—design for cross-zone communication.
9. Use **Azure Traffic Manager** for multi-region failover.
10. Separate **prod vs non-prod workloads** into different zones for isolation.

13. Azure Load Balancer

1. What is Azure Load Balancer?

A Layer 4 (TCP/UDP) load balancer that distributes network traffic across multiple VMs or services to ensure **high availability and scalability**.

2. What are the types of Azure Load Balancer?

- **Public Load Balancer** – Internet-facing, distributes external traffic.
- **Internal Load Balancer** – Distributes traffic within a VNet.

3. What is the difference between Basic and Standard Load Balancer?

Feature	Basic	Standard
SLA	None	99.99%
Scale	Limited	Unlimited
Availability Zones	Not supported	Supported
Security	Open by default	NSG by default

4. How does Azure Load Balancer determine which VM gets traffic?

- Uses a **hash-based distribution algorithm** (5-tuple hash: source IP, source port, destination IP, destination port, protocol).
- Ensures session persistence if configured.

5. What are Health Probes in Load Balancer?

Health probes check the status of backend VMs.

- If a VM fails the probe, the LB stops sending traffic to it.

6. Can Azure Load Balancer handle HTTPS traffic?

No. It's a **Layer 4 balancer**, so it does not terminate SSL. For HTTPS, use **Application Gateway or Front Door**.

7. What is NAT in Load Balancer?

- **Inbound NAT Rule:** Allows RDP/SSH to a specific VM instance.
- Example: Port 50001 → VM1:3389 (RDP).

8. How does Load Balancer handle outbound traffic?

- Provides **SNAT (Source Network Address Translation)** so backend VMs can access the internet using Load Balancer's public IP.

9. How does cross-zone load balancing work?

- In Standard SKU, traffic can be distributed across VMs in multiple Availability Zones.

10. Can you attach multiple frontends to a single Load Balancer?

Yes. A single load balancer can have **multiple frontend IPs** with different rules.

11. Difference between Load Balancer and Application Gateway?

Feature	Load Balancer	Application Gateway
Layer	4 (TCP/UDP)	7 (HTTP/HTTPS)
SSL Termination	❌	✅
Path-based Routing	❌	✅
Best For	General network traffic	Web applications

12. Difference between Load Balancer and Traffic Manager?

- **Load Balancer:** Distributes traffic within a region.
- **Traffic Manager:** DNS-based global traffic distribution across regions.

13. When should you use Load Balancer vs Front Door?

- **Load Balancer:** Low-level, region-specific, TCP/UDP traffic.
- **Front Door:** Global, web acceleration, SSL termination, WAF.

14. Can Load Balancer be used with VM Scale Sets?

Yes. It automatically integrates and balances traffic across scaled instances.

15. Is Load Balancer free?

- **Basic SKU:** Free.
- **Standard SKU:** Paid, based on rules and data processed.

16. How do you secure an Azure Load Balancer?

- Apply **NSGs** to backend pool.
- Use **Standard SKU** (secured by default).
- Prefer **Azure Bastion** over NAT for RDP/SSH.

17. What is HA Ports in Load Balancer?

- Allows a rule to load balance **all ports and protocols** on a VM.
- Useful for complex workloads (firewalls, VPN gateways).

18. How does session persistence work in Load Balancer?

Options:

- None (random distribution).
- Client IP (same client → same backend VM).
- Client IP + Protocol.

19. What metrics can you monitor in Azure Load Balancer?

- Data Path Availability (%)
- Health Probe Status

- SNAT Port Utilization
- Packet Drop Count

20. Best strategy: When to use Load Balancer vs Application Gateway vs Traffic Manager?

- **Load Balancer:** Internal/external, TCP/UDP traffic.
- **Application Gateway:** Web apps, Layer 7 (path routing, SSL).
- **Traffic Manager:** Global failover & DNS-based routing.

  👉 Often used **together** in multi-tier architectures.

Key Features

1. **Layer 4 (TCP/UDP) Load Balancing** – Distributes traffic based on transport-layer protocols.
2. **Public & Internal Load Balancer** – External (internet-facing) or internal (VNet-only).
3. **High Availability & Scalability** – Automatically scales to meet traffic demands.
4. **Inbound & Outbound Traffic Support** – Load balances both incoming requests and outbound connections (SNAT).
5. **Health Probes** – Detect unhealthy instances and stop routing traffic to them.
6. **Cross-Zone Load Balancing** – Spreads traffic across Availability Zones.
7. **NAT Rules** – Provides direct RDP/SSH access to specific VM instances.
8. **Multiple Frontends** – One load balancer can serve multiple IP addresses/ports.
9. **Integration with VM Scale Sets** – Automatically balances traffic to scaled-out instances.
10. **Free for Basic, Paid for Standard SKU** – Standard SKU adds HA, more features, and security.

Common Use Cases

1. **Web Applications** – Distribute HTTP/HTTPS traffic across multiple web servers.
2. **Gaming & Media Apps** – Handle large volumes of UDP traffic.
3. **High Availability for APIs** – Ensure uptime by routing to healthy instances.
4. **NAT Rules for Management** – Provide remote access to backend VMs securely.
5. **Database Load Balancing** – Balance SQL/NoSQL workloads inside VNet.
6. **IoT Applications** – Scale real-time device traffic.
7. **Outbound SNAT** – Centralized outbound internet access for VMs.

Best Practices

1. Always use **Standard SKU** for production (more secure + supports AZ redundancy).
2. Place backend VMs in an **Availability Set or Scale Set** for resiliency.
3. Configure **health probes** properly to detect real failures.
4. Use **zonal frontends** for zone resiliency.
5. For HTTPS, use **Application Gateway or Azure Front Door** instead of Load Balancer (Layer 7).
6. Monitor using **Azure Monitor metrics** (probe status, dropped packets).
7. Minimize NAT rules; prefer **Bastion** for secure RDP/SSH.
8. Combine with **Traffic Manager** for global load balancing.
9. Use **multiple frontends** to separate traffic types.
10. Apply **Network Security Groups (NSGs)** for backend security.

14. Azure Application Gateway

1. What is Azure Application Gateway?
A **Layer 7 load balancer** that routes and manages web traffic (HTTP/HTTPS) with advanced features like WAF, SSL termination, and path-based routing.

2. How is Application Gateway different from Azure Load Balancer?

Feature	Application Gateway	Load Balancer
OSI Layer	Layer 7 (HTTP/HTTPS)	Layer 4 (TCP/UDP)
SSL Termination	✅	❌
Path Routing	✅	❌
WAF	✅	❌
Best for	Web applications	General TCP/UDP traffic

3. What is the role of WAF in Application Gateway?
The **Web Application Firewall (WAF)** protects applications against common attacks like SQL injection, XSS, and OWASP Top 10 vulnerabilities.

4. What is SSL termination in Application Gateway?
The process of decrypting HTTPS traffic at the gateway before sending it to backend servers — reduces backend CPU load.

5. Can Application Gateway do end-to-end SSL?
Yes, it can re-encrypt traffic after termination before forwarding it to backend servers.

6. What is path-based routing in Application Gateway?
Allows requests like:

- /images/* → Image server pool
- /api/* → API backend
- /checkout/* → Payment system

7. What is multi-site hosting in Application Gateway?
Allows hosting multiple domains (e.g., abc.com, xyz.com) on the same Application Gateway instance with different listeners.

8. What are listeners in Application Gateway?
Listeners are rules that define how the gateway should accept traffic (e.g., HTTPS on port 443 for xyz.com).

9. What is the difference between Basic, Standard, and WAF SKUs?

- **Basic/Standard:** Load balancing + routing
- **WAF SKU:** Adds Web Application Firewall for security

10. Can Application Gateway work with on-premises servers?
Yes, by connecting via VPN/ExpressRoute and routing traffic to on-premises IPs.

11. How does autoscaling work in Application Gateway?
The gateway automatically scales instances up/down based on traffic load (Standard_v2 and WAF_v2 SKUs).

12. How does cookie-based session affinity work?
Ensures a user's session always routes to the same backend server using a session cookie.

13. What are connection draining and its use?
Gracefully removes unhealthy instances by allowing existing sessions to complete before stopping traffic.

14. What are redirection rules in Application Gateway?
Rules that allow redirection:

- HTTP → HTTPS
- One URL path → Another path
- One domain → Another domain

15. What diagnostic features does Application Gateway provide?

- Access Logs
- Performance Logs
- Firewall Logs
- Integration with Azure Monitor & Log Analytics

16. How does Application Gateway integrate with Azure Security?

- Works with **NSGs**
- WAF policies for security
- Can use **Private IP** for internal apps

17. When would you use Application Gateway over Front Door?

- **Application Gateway:** Regional web traffic routing + WAF
- **Front Door:** Global traffic routing + CDN + acceleration

18. Can Application Gateway integrate with Azure Kubernetes Service (AKS)?
Yes. Application Gateway can act as an ingress controller for AKS clusters.

19. What are common use cases for WAF mode in Application Gateway?

- Preventing SQL injection, XSS attacks
- Blocking malicious bots
- Protecting e-commerce checkout pages

20. How do you monitor Application Gateway?

- **Azure Monitor Metrics** (CPU, throughput, requests/sec)
- **WAF Logs** (blocked requests, attack signatures)
- **Log Analytics Queries** (troubleshooting & trend analysis)

Key Features

1. **Layer 7 Load Balancing** – Works at the HTTP/HTTPS layer.
2. **SSL Termination** – Decrypts HTTPS at the gateway, reducing VM load.
3. **Web Application Firewall (WAF)** – Protects apps from OWASP Top 10 threats (SQLi, XSS, etc.).
4. **Path-based Routing** – Route traffic to different backends based on URL path.
5. **Multi-site Hosting** – Host multiple web apps behind one gateway.
6. **Redirection Support** – HTTP → HTTPS redirection.
7. **Cookie-based Session Affinity** – Ensures user session stays on the same server.
8. **Autoscaling** – Automatically scales instances based on traffic.
9. **End-to-End SSL** – Option to re-encrypt traffic to backend servers.
10. **Integration with Azure Monitor** – Deep diagnostics and logging.

Common Use Cases

1. **Secure Web Applications** – Protect websites with WAF + SSL termination.
2. **Multi-Tenant Hosting** – Route multiple domains (e.g., abc.com, xyz.com) to different backend pools.
3. **API Gateway** – Route APIs to different microservices.
4. **E-commerce** – Direct /payment requests to secure, high-performance servers.
5. **Hybrid Cloud** – Route traffic to on-prem or Azure-hosted apps.
6. **HTTPS Redirection** – Force all traffic to secure connections.
7. **Global Web Applications** – Combined with Traffic Manager for worldwide distribution.

Best Practices

1. Always use **WAF SKU** for internet-facing applications.
2. Enable **HTTPS end-to-end** if backend security is critical.
3. Use **path-based routing** for microservices/web apps separation.
4. Combine with **Azure Front Door** for global apps requiring CDN + acceleration.
5. Configure **custom error pages** for user-friendly fallback.
6. Log all requests using **Azure Monitor + Log Analytics**.
7. Use **autoscaling** for unpredictable workloads.
8. Keep **listener configuration simple** (avoid too many rules per gateway).
9. Secure backends with **NSGs + Private IPs**.
10. Monitor **WAF logs** regularly for suspicious activity.

15. Azure Traffic Manager

1. What is Azure Traffic Manager?
A **DNS-based global traffic distribution service** that directs client requests to the best available endpoint based on routing rules and health checks.

2. How does Traffic Manager differ from Azure Load Balancer?

- **Traffic Manager:** DNS-based, global, routes at **DNS level**.
- **Load Balancer:** Layer 4, regional, routes at **network packet level**.

3. How does Traffic Manager differ from Azure Front Door?

Feature	Traffic Manager	Front Door
Routing	DNS-based	Proxy-based (Layer 7)
Latency	Depends on DNS cache	Real-time
WAF/SSL Offload	❌	✅
Best For	Global DNS routing	Web acceleration & security

5. **Does Traffic Manager proxy or inspect traffic?**
No. It only returns DNS records; the traffic flows directly to endpoints.

5. What types of endpoints can Traffic Manager support?

- Azure VMs, Web Apps, Cloud Services
- External (non-Azure) endpoints
- Nested Traffic Manager profiles

6. What are the routing methods supported by Traffic Manager?

- **Priority (Failover):** Routes to primary, fails over to backup.
- **Weighted:** Distributes traffic based on weights (A/B testing).
- **Performance:** Routes to the lowest latency endpoint.
- **Geographic:** Routes based on user's region.
- **Multivalue:** Returns multiple healthy endpoints.
- **Subnet:** Routes based on user IP ranges.

7. Which routing method would you use for disaster recovery?
👉 **Priority routing** (primary → secondary).

8. Which routing method is best for load testing new versions?
👉 **Weighted routing** (e.g., 90% old, 10% new).

9. Which routing method ensures lowest latency for users?
👉 **Performance routing**.

10. Which routing method ensures GDPR/data residency compliance?
👉 **Geographic routing**.

11. How does Traffic Manager check endpoint health?
Uses **HTTP, HTTPS, or TCP health probes** at regular intervals.

12. What happens if all endpoints fail health checks?
Traffic Manager returns **all endpoints** (including unhealthy) so clients may retry.

13. What is the role of TTL in Traffic Manager?
TTL (Time-to-Live) determines how long DNS responses are cached. Lower TTL = faster failover but more DNS queries.

14. Can Traffic Manager route traffic between Azure and non-Azure endpoints?
Yes. It supports **external endpoints** (on-prem or other clouds).

15. What is a nested Traffic Manager profile?
A configuration where one profile points to another profile → enables advanced scenarios (multi-tier failover + performance).

16. How does Traffic Manager integrate with Azure Monitor?

- Alerts on endpoint health
- Latency tracking
- Failover events

17. Can Traffic Manager be used with Azure Front Door?
Yes. Common pattern:

- **Traffic Manager:** Global failover (DNS).
- **Front Door:** Application acceleration + WAF.

18. Can Traffic Manager route UDP traffic?
Indirectly. Since it is DNS-based, it works with any protocol (TCP, UDP, HTTP, HTTPS, etc.).

19. Is Traffic Manager free?
No. Pricing is based on:

- Number of DNS queries
- Number of health checks

20. How would you design a highly available global architecture with Traffic Manager?

- Deploy app in multiple Azure regions
- Configure Traffic Manager with **priority routing** (for DR) + **performance routing** (for latency)
- Enable **low TTL** for fast failover
- Combine with **Front Door or App Gateway** for Layer 7 features

Key Features

1. **DNS-based Load Balancing** – Routes traffic using DNS, not a proxy.
2. **Global Traffic Distribution** – Directs users to the nearest/healthiest endpoint.
3. **Multiple Routing Methods** – Priority, Weighted, Performance, Geographic, Multivalue, Subnet.
4. **Failover Support** – Automatically reroutes traffic if the primary endpoint is unavailable.
5. **Supports Azure + External Endpoints** – Can route traffic to Azure services or non-Azure endpoints.
6. **Health Probes** – Monitors endpoint availability via HTTP, HTTPS, or TCP.
7. **Scalable & Highly Available** – Global service with built-in redundancy.
8. **Hybrid Cloud Support** – Works across Azure, on-prem, and other clouds.
9. **Customizable TTL** – Controls how quickly DNS changes propagate.
10. **Integration with Azure Monitor** – For alerting and diagnostics.

Common Use Cases

1. **Global Website Distribution** – Send users to the closest data center.
2. **Disaster Recovery / Failover** – Switch to backup region if primary fails.
3. **Latency Optimization** – Route users to the lowest-latency region.
4. **Multi-Region Deployment** – Balance workloads across continents.
5. **A/B Testing or Gradual Rollout** – Weighted routing to test new versions.
6. **Regulatory Compliance** – Route users based on geography (e.g., EU users stay in EU).
7. **Hybrid Cloud Apps** – Balance traffic between Azure and on-prem resources.
8. **Enterprise Applications** – Provide high availability for mission-critical workloads.

Best Practices

1. Use **low TTL** for faster failover (e.g., 30s–60s).
2. Always configure **multiple endpoints** for redundancy.
3. Combine with **Azure Front Door** for Layer 7 acceleration + WAF.
4. Use **priority routing** for disaster recovery scenarios.
5. Apply **geographic routing** for data residency compliance.
6. Monitor **endpoint health probes** closely.
7. Integrate with **Application Insights** for latency monitoring.
8. Avoid over-reliance on DNS caching – use **short TTLs** for dynamic workloads.
9. Secure endpoints with **SSL/TLS** even though ATM doesn't proxy traffic.
10. Use **Traffic View** (ATM feature) to analyze user traffic patterns.

16. Azure Firewall

1. What is Azure Firewall?
A **managed, cloud-native, stateful firewall service** that provides network and application-level protection for Azure resources.

2. How is Azure Firewall different from NSG (Network Security Group)?

- **NSG:** Simple packet filtering (Layer 3/4).
- **Firewall:** Stateful inspection, application rules, FQDN filtering, logging, threat intelligence.

3. What is the difference between Azure Firewall and Azure Application Gateway WAF?

Feature	Azure Firewall	App Gateway WAF
Layer	3–7	7 (HTTP/HTTPS only)
Scope	Network-wide	Web apps
Use Case	Protect VNets, outbound/inbound	Protect websites from OWASP attacks

4. What are Firewall Policies?
Centralized rule collections that define **application, network, DNAT, and threat intelligence** rules, reusable across multiple firewalls.

5. What is stateful inspection in Azure Firewall?
It keeps track of **active sessions** and ensures only valid packets in an existing connection are allowed.

6. What are Application Rules in Azure Firewall?
Rules that allow/deny **HTTP/S traffic** based on FQDN, domains, or categories (e.g., allow *.microsoft.com).

7. What are Network Rules in Azure Firewall?
Rules that filter traffic based on **IP addresses, ports, and protocols** (TCP, UDP, ICMP).

8. What is DNAT in Azure Firewall?
Destination NAT – maps **public IPs** to **private internal resources**, e.g., allow RDP access from the internet.

9. What is SNAT in Azure Firewall?
Source NAT – changes the source IP of outbound traffic to firewall's public IP for internet access.

10. Can Azure Firewall filter traffic based on URLs?
Yes (via Application Rules + FQDN filtering).

11. What is Threat Intelligence in Azure Firewall?
Feature that blocks or alerts on traffic from/to **known malicious IPs/domains**.

12. How do you secure RDP/SSH using Azure Firewall?

- Use **DNAT rules** sparingly.
- Best practice: use **Azure Bastion** instead.

13. Can Azure Firewall work across multiple VNets?
Yes, using **Hub-and-Spoke** architecture with VNet peering or Virtual WAN.

14. How does Azure Firewall integrate with SIEM tools?
It sends logs to **Azure Monitor / Log Analytics / Event Hub / Sentinel** for monitoring & analysis.

15. What are Firewall Availability Zones?
Deployment option where Firewall instances are distributed across **AZs** for resiliency.

16. When would you use Azure Firewall vs NSG vs WAF?

- **Firewall:** Network-wide, outbound/inbound filtering.
- **NSG:** Simple packet filtering within VNets.
- **WAF:** Protects **web apps** (Layer 7 attacks).

17. Can Azure Firewall filter outbound traffic to malicious websites?
Yes, using **application rules + threat intelligence**.

18. How does Azure Firewall scale?
It's **elastic** – scales automatically based on traffic, no manual configuration needed.

19. Can Azure Firewall be deployed in on-prem environments?
No, it's **Azure-only**. For hybrid, traffic must flow via Azure (VPN/ExpressRoute).

20. How would you design a secure hub-and-spoke network with Azure Firewall?

- Place **Azure Firewall in Hub VNet**.
- Route **all spoke VNets traffic** through hub (UDR).
- Apply **centralized firewall policy**.
- Use **NSGs on subnets + Firewall for advanced filtering**.

Key Features

1. **Fully Managed Firewall-as-a-Service** – Cloud-native, highly available, scales automatically.
2. **Stateful Firewall** – Keeps track of active connections (TCP/UDP).
3. **Application Rules** – Allow/Deny traffic based on FQDN, URL, or category.
4. **Network Rules** – Control traffic by IP addresses, ports, and protocols.
5. **Threat Intelligence** – Blocks traffic from/to known malicious IPs/domains.
6. **Built-in High Availability** – No need to configure load balancers.
7. **Outbound SNAT** – Provides outbound internet connectivity using firewall's public IP.
8. **Inbound DNAT** – Maps public IPs to private resources.
9. **Integration with Azure Monitor & Sentinel** – Centralized logging & SIEM.
10. **Support for Forced Tunneling** – Send all traffic to on-prem or another security appliance.

Common Use Cases

1. **Centralized Security Control** – Protects all workloads in a VNet or multiple VNets.
2. **Application Filtering** – Control web access for VMs based on domain names.
3. **Hybrid Connectivity** – Secure traffic between Azure and on-premises networks.
4. **Internet Outbound Protection** – Prevents VMs from accessing malicious websites.
5. **Inbound DNAT Protection** – Securely expose specific services (e.g., RDP/SSH).
6. **Multi-VNet Hub-and-Spoke Architecture** – Firewall in hub VNet, spoke VNets route through it.
7. **Compliance** – Enforce security policies across regulated industries.
8. **Threat Detection** – Stop botnets and malware connections using threat intelligence.

Best Practices

1. Deploy Azure Firewall in a **Hub-and-Spoke architecture** for centralized control.
2. Use **Firewall Policy** (instead of classic rules) for easier management.
3. Enable **Threat Intelligence Mode = Alert + Deny** for stronger security.
4. Log all traffic to **Log Analytics / Sentinel** for auditing.
5. Use **DNAT rules** sparingly (minimize public exposure).
6. Combine with **Azure Bastion** instead of exposing RDP/SSH.
7. Apply **Application Rules** for outbound web filtering (instead of broad NSG rules).
8. Use **Availability Zones** for zonal resiliency.
9. Enable **forced tunneling** for compliance.
10. Regularly review rules to avoid overly permissive access.

17. Azure VPN Gateway

1. What is Azure VPN Gateway?
A **networking service** that provides secure, encrypted connectivity between Azure VNets and on-premises networks or remote clients using VPN protocols.

2. What are the main types of VPN connections in Azure?
- **Site-to-Site (S2S):** Connects on-premises datacenter to Azure.
- **Point-to-Site (P2S):** Individual devices connect remotely.
- **VNet-to-VNet:** Connects multiple VNets together.

3. How does VPN Gateway differ from ExpressRoute?
- **VPN Gateway:** Uses public internet (encrypted IPsec tunnels).
- **ExpressRoute:** Private dedicated connection, faster & more reliable.

  👉 They can be used together for failover.

4. What protocols does Azure VPN Gateway support?
- **IKEv2/IPsec** (most common)
- **SSTP** (for Windows clients)
- **OpenVPN** (cross-platform P2S)

5. What is the Gateway Subnet in Azure VPN Gateway?
A dedicated subnet (named GatewaySubnet) that hosts the VPN Gateway resources. Must be at least /27.

6. What are VPN Gateway SKUs?
Performance-based tiers (e.g., VpnGw1–VpnGw5). Higher SKUs = more throughput + connections.

7. What is Active-Active mode in VPN Gateway?
Deploys **two gateway instances** in active mode for HA, each with its own public IP.

8. How does BGP help in VPN Gateway?
BGP enables **dynamic route exchange** between Azure and on-premises networks, reducing manual configuration.

9. Can VPN Gateway connect multiple on-premises sites?
Yes. Multiple S2S tunnels can be established to connect branch offices.

10. Can VPN Gateway connect across Azure regions?
Yes. VNet-to-VNet VPN can connect across different regions.

11. How is traffic secured in VPN Gateway?
Traffic is encrypted using **IPsec/IKE** standards (AES256, SHA256, etc.).

12. What is the difference between Policy-based and Route-based VPNs?

- **Policy-based:** Static rules, less flexible.
- **Route-based:** Uses routing tables, supports BGP, recommended.

13. What's the max throughput for VPN Gateway?
Depends on SKU – up to **10 Gbps (VpnGw5)**.

14. How does Point-to-Site VPN authenticate users?

- **Certificates** (self-signed or CA)
- **Azure AD authentication** (modern approach)

15. What happens if the VPN Gateway fails?

- In **Active-Standby**: Failover occurs, short downtime.
- In **Active-Active**: Seamless failover with two tunnels.

16. When should you use VPN Gateway instead of ExpressRoute?

- Small/medium workloads
- Temporary or dev/test environments
- Cost-sensitive scenarios

17. Can you use VPN Gateway and ExpressRoute together?
Yes. Common setup: ExpressRoute for primary connectivity, VPN Gateway as failover.

18. How do you troubleshoot VPN Gateway connection issues?

- Check **Azure Monitor diagnostic logs**.
- Verify **shared keys** for S2S VPN.
- Ensure **IP ranges don't overlap**.
- Run **Azure Network Watcher VPN Troubleshoot** tool.

19. Can VPN Gateway be used for cross-tenant connectivity?
Yes, using VNet-to-VNet or S2S tunnels across subscriptions/tenants.

20. How would you design a highly available hybrid network with VPN Gateway?

- Deploy VPN Gateway in **Active-Active mode**.
- Enable **Zone-Redundant Gateway**.
- Use **ExpressRoute + VPN Gateway failover**.
- Configure **BGP for dynamic routing**.
- Use **Hub-and-Spoke architecture** for centralized connectivity

Key Features

1. **Site-to-Site VPN (S2S)** – Securely connects on-premises networks to Azure VNets.
2. **Point-to-Site VPN (P2S)** – Allows individual devices (remote users) to connect securely.
3. **VNet-to-VNet VPN** – Securely connects multiple VNets (same or different regions).
4. **Supports Multiple VPN Protocols** – IKEv2, IPsec, SSTP, and OpenVPN.
5. **High Availability Options** – Active-Active mode for redundancy.
6. **Zone-Redundant Gateways** – Availability Zone support for resiliency.
7. **ExpressRoute Integration** – Can be combined with ExpressRoute for failover.
8. **Custom Routing Support** – Configurable with UDRs and BGP (Border Gateway Protocol).
9. **Scalability with SKUs** – Different SKUs available (VpnGw1, VpnGw2, VpnGw3, etc.).
10. **Security** – Data is encrypted using IPsec/IKE standards.

Common Use Cases

1. **Hybrid Cloud** – Connect on-premises datacenter to Azure securely.
2. **Remote Workforce** – Employees connect via Point-to-Site VPN.
3. **Multi-Region Apps** – VNet-to-VNet connections for redundancy.
4. **Disaster Recovery** – Failover traffic from on-prem to Azure.
5. **Development/Testing** – Temporary secure access for developers.
6. **Secure Data Transfers** – Encrypted communication for compliance.
7. **Backup Connectivity** – As a backup when ExpressRoute goes down.
8. **Partner Connectivity** – Allow vendors/partners secure network access.

Best Practices

1. Choose the **right SKU** (VpnGw1–5, depending on throughput and connections).
2. Use **Active-Active configuration** for HA and redundancy.
3. Prefer **IKEv2** for stronger security and better compatibility.
4. Use **BGP** for dynamic routing instead of static routes.
5. For mission-critical apps, use **ExpressRoute with VPN failover**.
6. Always enable **Azure Monitor logs** for troubleshooting.
7. Use **Gateway Subnet** with recommended size (/27 or larger).
8. Avoid overlapping IP ranges between on-prem and Azure VNets.
9. Enable **Zone Redundancy** for higher availability.
10. Secure **Point-to-Site VPNs** with certificates or Azure AD authentication.

18. Azure DNS

1. What is Azure DNS?
A cloud-based service for hosting DNS zones and records, providing fast, reliable, and scalable domain name resolution.

2. What are the two types of DNS zones in Azure?

- **Public DNS Zone** → For internet-facing domains.
- **Private DNS Zone** → For internal name resolution within VNets.

3. What DNS record types are supported in Azure DNS?
A, AAAA, CNAME, MX, TXT, NS, SOA, PTR, SRV, Alias records.

4. What is an Alias record in Azure DNS?
A special record type that maps a domain directly to an Azure resource (Traffic Manager, App Service, CDN, etc.), automatically updating if the resource changes.

5. What's the difference between Azure DNS and Azure Private DNS?

- **Azure DNS (Public)** → Resolves names for internet users.
- **Private DNS** → Resolves names for Azure VNets without exposing them publicly.

6. How does Azure DNS ensure high availability?
It uses Azure's global anycast network with redundant DNS servers worldwide.

7. How can DNS be secured in Azure?

- Use **RBAC** for access control.
- Enable **audit logs**.
- Lock DNS zones to prevent accidental deletion.

8. How is DNS query performance optimized in Azure?

- Anycast-based global distribution.
- Caching with DNS resolvers.
- Configurable TTL values.

9. Can Azure DNS be used as a registrar?
No. Azure DNS only hosts domains; you must purchase/register domains via another registrar and then delegate them to Azure DNS.

10. How do you enable split-horizon DNS in Azure?
By using **both Public and Private DNS zones** for the same domain name.

11. How does Azure DNS integrate with VNets?
Through **Private DNS Zones**, which automatically link to VNets for internal name resolution.

12. What is auto-registration in Azure Private DNS?
When enabled, Azure automatically registers VM hostnames into the Private DNS Zone when VMs are created or deleted.

13. How do you manage DNS in Azure using automation?

- Azure CLI, PowerShell, REST API, ARM/Bicep/Terraform templates.

14. How does Azure DNS integrate with Traffic Manager?
You can create **Alias records** that point a DNS name to a Traffic Manager profile for global load balancing.

15. Can Azure DNS resolve on-prem names in hybrid setups?
Yes, using **conditional forwarding** and **Private DNS Zones linked with VNets + VPN/ExpressRoute**.

16. How would you configure DNS for a multi-region app in Azure?

- Use **Azure Traffic Manager** with DNS for geo-routing.
- Create DNS alias records for failover.

17. How does Azure DNS support email validation?
By hosting **MX, SPF, DKIM, and DMARC TXT records**.

18. How do you delegate a domain to Azure DNS?
Update the **NS records** at your domain registrar to point to the Azure DNS name servers.

19. How can DNS changes be monitored in Azure?

- Enable **Azure Monitor + Diagnostic Logs** for DNS zones.
- Use **Azure Activity Log** for change auditing.

20. Real-world scenario: Your app must support both internal and external resolution. How do you design DNS?

- Use **Public DNS Zone** for internet access.
- Use **Private DNS Zone** linked to VNets for internal workloads.
- Implement **split-horizon DNS** for the same domain name.

Key Features

1. **Global DNS Hosting** – Host your DNS domains in Azure.
2. **Private and Public Zones** – Supports both internet-facing (public) and internal (private) DNS.
3. **High Availability** – Built on Azure's global infrastructure with SLA-backed uptime.
4. **Scalability** – Handles millions of queries per second.
5. **Fast Performance** – Low-latency name resolution using Azure's global anycast network.
6. **Custom DNS Records** – A, AAAA, CNAME, MX, TXT, NS, PTR, SRV, and SOA supported.

7. **DNS Aliases (Alias Records)** – Point root domain (@) to Azure resources like Traffic Manager, App Service, or CDN.
8. **Secure Management** – Integrated with Azure RBAC and logging.
9. **Automation** – Full REST API, ARM templates, PowerShell, CLI support.
10. **Integration** – Works with Azure services like VNet, Traffic Manager, CDN, Application Gateway.

Common Use Cases

1. Hosting custom domain names in Azure (e.g., mycompany.com).
2. Internal DNS resolution in **private VNets** without using external DNS.
3. Pointing root domains (@) directly to Azure services using **alias records**.
4. Hybrid scenarios where on-prem and Azure resources need shared DNS resolution.
5. High-performance global DNS resolution with **Traffic Manager + DNS**.
6. Custom domain validation for SSL/TLS certificates.
7. DNS-based email validation (SPF, DKIM, DMARC records).
8. Multi-cloud DNS management using Azure as primary or secondary.
9. Failover routing with **alias records + Traffic Manager**.
10. Logging & monitoring DNS queries for compliance/security.

Best Practices

1. Use **Alias records** for Azure services instead of IP addresses.
2. Implement **role-based access control (RBAC)** for DNS zone modifications.
3. Use **Azure Private DNS Zones** for internal workloads instead of custom DNS servers.
4. Enable **logging and auditing** for DNS changes.
5. Keep **TTL values short** for frequently changing services.
6. Design **redundant DNS architectures** (Azure DNS + third-party DNS if needed).
7. For hybrid networks, integrate **Azure DNS Private Zones with on-prem DNS** using conditional forwarding.
8. Regularly monitor DNS query analytics for anomalies.
9. Secure DNS zones with **least-privilege access**.
10. Use **infrastructure as code (ARM/Bicep/Terraform)** to deploy DNS consistently.

19. Azure Backup, Recovery Service Vault

1. What is Azure Backup?
A cloud-based service that provides simple, secure, and cost-effective data backup and recovery for Azure and on-prem workloads.

2. What is a Recovery Services Vault (RSV)?
A **logical storage container** in Azure that holds backup data, recovery points, and configurations.

3. Difference between Backup Vault and Recovery Services Vault?

- **Backup Vault** → Legacy, supports only specific workloads.
- **RSV** → Modern, supports **Azure Backup + Azure Site Recovery** with broader functionality.

4. What storage options are available for RSV?

- **Locally Redundant Storage (LRS)**
- **Geo-Redundant Storage (GRS)**

5. What types of workloads can Azure Backup protect?

- Azure VMs
- SQL in Azure VMs
- SAP HANA in Azure VMs
- Files & Folders (via MARS agent)
- On-prem servers (via MABS/DPM)

6. What is Soft Delete in Azure Backup?
A feature that retains deleted backups for **14 days** to protect against accidental or malicious deletions.

7. Can you extend retention beyond 14 days?
Yes, using **Immutable Vault** with configurable retention policies.

8. How are backups secured in Azure?

- Data encrypted in transit (TLS).
- Data encrypted at rest (Azure Storage Encryption).
- Optional customer-managed keys in Key Vault.

9. What is an Azure Backup Policy?
A schedule and retention rule defining when backups run and how long they are stored.

10. How do you prevent unauthorized backup deletion?

- Enable **Soft Delete**.
- Use **RBAC + MUA (Multi-User Authorization)**.

11. Can Azure Backup integrate with on-prem systems?
Yes, via **MARS agent, MABS, or System Center DPM**.

12. What's the difference between Azure Backup Agent (MARS) and Azure Backup Server (MABS)?

- **MARS agent** → Files/folders only.
- **MABS** → Broader workloads (SQL, Hyper-V, VMware).

13. How is monitoring done for Azure Backup?

- Azure Portal (Vault view).
- **Azure Monitor** alerts.
- **Log Analytics** queries.

14. How does Azure Backup support compliance?
By providing **immutable storage**, **long-term retention**, and **audit logs** for regulatory compliance.

15. What is Instant Restore in Azure Backup?
Allows VM recovery quickly using **snapshots** before full backup transfer completes.

16. How do you handle backup costs?

- Use **LRS for dev/test**, GRS for production.
- Move older backups to **archive tier**.
- Optimize backup frequency & retention.

17. What's the difference between Azure Backup and Azure Site Recovery (ASR)?

- **Backup** → Protects data & workloads for restore.
- **ASR** → Provides disaster recovery with failover/failback.

18. Can RSVs be moved across regions?
No, RSVs are region-bound. You must create a new vault and reconfigure.

19. How do you restore a single file from an Azure VM backup?
Use **File Recovery** option in the vault → Mounts a recovery point as a temporary drive.

20. Real-world scenario: A ransomware attack encrypted files in Azure VM. How does Azure Backup help?

- Use **point-in-time restore** from a clean backup.
- Leverage **Soft Delete** if backups were tampered with.
- Ensure MFA + RBAC prevented malicious deletion.

Key Features

1. **Centralized Backup Management** – Manage all backups (VMs, databases, files, workloads) from one place.
2. **Wide Workload Support** – Protects Azure VMs, SQL databases, on-prem servers, SAP HANA, and more.
3. **Recovery Services Vault (RSV)** – A storage container in Azure for backup and disaster recovery data.
4. **Policy-Based Automation** – Schedule and enforce backup policies automatically.
5. **Incremental Backups** – Only changes are backed up, saving cost and time.
6. **Geo-Redundant Storage (GRS)** – Provides durability by replicating backup data to a secondary region.
7. **Long-Term Retention** – Store backups for years (up to 99 years).
8. **Encryption at Rest & Transit** – Data is secured with Azure Storage encryption + TLS.
9. **Instant Restore & File Recovery** – Recover files, folders, or entire workloads.
10. **Integration with On-Prem** – Azure Backup Server (MABS) and System Center DPM support hybrid scenarios.

Common Use Cases

1. **Backup Azure VMs** – Full disk and OS-level backup for disaster recovery.
2. **SQL & SAP HANA Backup** – Protect business-critical databases.
3. **File & Folder Backup** – For Windows/Linux machines (on-prem and Azure).
4. **Long-Term Data Retention** – Meet compliance needs (HIPAA, ISO, GDPR).
5. **Disaster Recovery (DR)** – Combine with **Azure Site Recovery (ASR)** for failover scenarios.
6. **Hybrid Backup** – Protect both on-prem workloads and cloud resources.
7. **Ransomware Protection** – Backup data with multi-factor authentication (MFA) for deletion.
8. **Point-in-Time Recovery** – Restore workloads to a specific past state.
9. **Cost-Effective Archival** – Store backups in cool/archive tiers.
10. **Centralized Monitoring** – Track backup jobs using **Azure Monitor & Log Analytics**.

Best Practices

1. Always use a **Recovery Services Vault** (RSV) instead of classic Backup Vaults.
2. Enable **Soft Delete** (default 14 days) to recover accidentally deleted backups.
3. Choose **Geo-Redundant Storage (GRS)** for production workloads, **Locally Redundant Storage (LRS)** for dev/test.
4. Implement **role-based access control (RBAC)** to restrict backup management.
5. Set **backup policies** that align with RPO (Recovery Point Objective) and RTO (Recovery Time Objective).
6. Monitor backup jobs and configure **alerts** for failures.
7. Regularly **test restores** to validate backup reliability.
8. Enable **multi-user authorization (MUA)** for critical operations (like deleting backup vaults).
9. Optimize cost by archiving long-term backups to **Azure Backup Vault tiering**.
10. Integrate with **Azure Policy** for compliance enforcement.

20. Azure App Service & Azure App Service plan

1. What is Azure App Service?
A fully managed **Platform as a Service (PaaS)** for hosting web apps, REST APIs, and mobile backends.

2. What is an App Service Plan?
It defines the **compute resources (region, VM size, pricing tier)** used by one or more App Service apps.

3. What languages and runtimes do App Service support?
.NET, Java, Node.js, Python, PHP, Ruby, Go, and custom containers.

4. What's the difference between Free, Shared, Basic, Standard, and Premium tiers?

- **Free/Shared** → Test/dev, shared infra, no SLA.
- **Basic** → Dedicated VMs, no autoscale.
- **Standard** → Dedicated, autoscale, SLA-backed.
- **Premium** → Advanced scaling, better performance, VNet integration.
- **Isolated (ASE)** → Highest performance, runs inside private VNet.

5. How does billing work for App Service?
Billing is based on the **App Service Plan**, not individual apps. Multiple apps can share a plan.

6. What are deployment slots in App Service?
Environments (staging, QA, prod) where you can deploy and swap apps with zero downtime.

7. How do you scale App Services?

- **Scale Up (Vertical)** → Increase VM size (more CPU/memory).
- **Scale Out (Horizontal)** → Increase number of instances.

8. What scaling options are available in App Service Plans?

- Manual scaling
- Autoscaling (CPU, memory, request count, schedule-based)

9. What is the difference between scaling up vs scaling out?

- **Scaling up** → More powerful VM (same instance).
- **Scaling out** → More instances running the same app.

10. Can multiple apps share one App Service Plan?
Yes, but they share the same compute resources.

11. How do you secure apps in App Service?

- Authentication with Microsoft Entra ID/other IdPs.
- Managed Identity for Azure resources.
- Private Endpoint or VNet integration.

12. What is VNet integration in App Service?
Allows an app to securely access resources inside an Azure VNet (databases, services).

13. What is Hybrid Connections?
Feature that enables App Service apps to securely connect to **on-prem resources**.

14. How does Managed Identity work in App Service?
Provides a service identity that apps use to authenticate to Azure resources without credentials.

15. What are App Service Environment (ASE) and Isolated Plans?
ASE runs App Services in a **fully isolated, dedicated VNet** for high security and compliance.

16. How do you perform blue-green deployments with App Service?
Use **deployment slots**, deploy to staging, then **swap** with production.

17. What monitoring tools integrate with App Service?

- Application Insights
- Azure Monitor
- Log Analytics

18. How does App Service handle custom domains and SSL?

- Supports domain mapping via DNS (CNAME, A records).
- Provides free App Service Managed Certificates or bring-your-own.

19. How can you reduce costs with App Service Plans?

- Consolidate apps into one plan (if usage is low).
- Use auto-scaling rules instead of overprovisioning.
- Choose **LRS storage + lower tiers** for dev/test apps.

20. Real-world scenario: You have 5 small apps. Should you use 5 App Service Plans?
No. You can host all 5 apps under **one App Service Plan** to save cost (unless they need different scaling or regions).

Key Features (Azure App Service)

1. **Fully Managed PaaS** – Host web apps, APIs, and mobile backends without managing infrastructure.
2. **Multi-Language Support** – .NET, Java, Node.js, Python, PHP, Ruby, Go, and containers.
3. **Multiple Deployment Options** – GitHub Actions, Azure DevOps, ZIP, Docker, FTP, Visual Studio.
4. **Auto-Scaling** – Scale up (increase resources) or out (increase instances).
5. **Custom Domains & SSL** – Bind custom domains with free/paid SSL certificates.
6. **Authentication & Authorization** – Built-in integration with Microsoft Entra ID, Google, Facebook, etc.
7. **Staging Slots** – Deploy apps in staging before swapping to production.
8. **Hybrid Networking** – Connect securely to VNets and on-prem systems.
9. **Integrated Monitoring** – Application Insights, Log Analytics, Azure Monitor.

10. **Global Availability** – Deploy apps across multiple Azure regions.

Key Features (Azure App Service Plan)

1. **Defines Compute Resources** – Specifies VM size, region, and pricing tier for App Service apps.
2. **Multiple Apps per Plan** – Host multiple web apps/APIs under one plan (shared resources).
3. **Pricing Tiers** – Free, Shared, Basic, Standard, Premium, Isolated (ASE).
4. **Scaling Support** – Manual and auto-scaling based on CPU, memory, HTTP queue, or schedules.
5. **Dedicated vs Shared** – Choose between shared infrastructure (cheap) or dedicated VMs (better performance).
6. **Region Binding** – All apps under a plan run in the same region.
7. **Integrated with ASE** – Premium/Isolated tiers can run inside **App Service Environment (ASE)**.
8. **Billing Model** – Pay for the **App Service Plan** (compute), not individual apps.
9. **High Availability** – 99.95% SLA when using Standard or higher tiers.
10. **Flexibility** – Change tiers or scale without redeploying apps.

Common Use Cases

1. Hosting **enterprise websites** with custom domains and SSL.
2. Deploying **RESTful APIs** for mobile/web applications.
3. Running **microservices** inside App Service (or containers).
4. Hosting **internal line-of-business apps** with VNet integration.
5. **Multi-region deployments** for global performance.
6. **Zero-downtime deployments** with deployment slots.
7. Running **multi-tenant SaaS apps** under a single plan.
8. Cost optimization by hosting multiple apps in one App Service Plan.
9. Secure **authentication integration** with Microsoft Entra ID.
10. **Event-driven scaling** of APIs with auto scale policies.

Best Practices

1. Choose the **right App Service Plan tier** based on workload (don't overprovision).
2. Use **deployment slots** for blue-green/canary deployments.
3. Enable **autoscaling** for production workloads.
4. Secure apps with **Managed Identity + App Service Authentication**.
5. Monitor apps with **Application Insights** for performance and errors.
6. Use **Azure Front Door or App Gateway** for global load balancing + WAF.
7. Keep apps in **separate plans** if they have different scaling/region needs.
8. Use **VNet Integration** for secure backend connectivity.
9. Automate deployments with **CI/CD pipelines** (GitHub Actions, Azure DevOps).
10. Regularly **review metrics & optimize scaling rules**.

21. Azure App Service Environment

1. What is Azure App Service Environment (ASE)?
ASE is a **fully isolated and dedicated hosting environment** for securely running Azure App Service apps inside a customer's **Azure VNET**.

2. What's the difference between App Service and App Service Environment?

- **App Service:** Multi-tenant, shared infrastructure.
- **ASE:** Single-tenant, isolated, runs inside your VNET.

3. What types of apps can you host in ASE?
Web Apps, API Apps, Mobile Apps, and Azure Functions.

4. What are the main versions of ASE?

- **ASEv2:** Complex setup, more expensive.
- **ASEv3:** Simplified deployment, better performance, reduced cost.

5. What is ILB ASE?
Internal Load Balancer ASE → An ASE configured with **private IPs only**, accessible within a VNET.

6. How does ASE integrate with a VNET?
ASE is deployed inside a **subnet of your VNET**, enabling private IP-based access control.

7. What is the minimum subnet size for ASE?
A **/27 subnet** or larger (recommended /24) to support scaling.

8. How do you secure inbound traffic to ASE?

- Internal ASE (ILB) → Private only.
- External ASE → Place behind **App Gateway + WAF**.

9. How does ASE handle outbound access?
Traffic flows through your VNET → controlled by **NSGs, firewalls, private endpoints**.

10. Can ASE scale across Availability Zones?
Yes, ASEv3 supports **Zone Redundancy** for higher resiliency.

11. How does ASE scale applications?

- **Scale-up** → Increase instance size.
- **Scale-out** → Add more instances across workers.

12. How many apps can ASE host?
Hundreds to thousands, depending on instance and SKU limits.

13. What pricing tier does ASE use?
Isolated tier (I1–I3, I4–I8v2) specifically designed for ASE.

14. What's the benefit of ASE vs Dedicated App Service Plan?
ASE provides **network isolation & VNET integration**, which a regular dedicated App Service Plan does not.

15. How is ASE different from AKS (Azure Kubernetes Service)?

- **ASE:** PaaS, managed hosting for apps, no container orchestration needed.
- **AKS:** Container orchestration platform, more control, but higher management effort.

16. When should you choose ASE over App Service?

- Regulatory compliance (finance, healthcare, government).
- Apps needing **private-only access**.
- Large-scale, high-security enterprise workloads.

17. How can you expose apps in ASE to the internet securely?
Deploy an **App Gateway + WAF** in front of the ASE.

18. Can ASE apps access on-prem resources?
Yes, via **VPN Gateway or ExpressRoute** integrated with the ASE's VNET.

19. How do you monitor ASE performance?

- **Azure Monitor & Metrics**
- **Application Insights**
- **Log Analytics workspace**

20. What are some limitations of ASE?

- Higher cost than shared App Service.
- Longer provisioning time (~1–2 hours).
- Requires large subnet planning.

Key Features

1. **Fully Isolated & Dedicated** – Runs App Services in a private, dedicated environment.
2. **High Scale** – Can host **hundreds of App Service instances**.
3. **VNET Integration** – Deploy inside your Azure Virtual Network (private IPs).
4. **Inbound & Outbound Network Control** – Control access with NSGs, firewalls, and private endpoints.
5. **Supports Multiple App Types** – Web Apps, API Apps, Mobile Apps, Functions.
6. **Scaling** – Scale up (larger instances) or scale out (more instances).
7. **Zone-Redundant ASEv3** – Higher resiliency with multi-AZ deployments.
8. **Enhanced Security** – Apps not exposed to the public internet unless configured.
9. **Custom Domains & SSL** – Fully supported in isolated environments.
10. **Enterprise Workloads** – Designed for compliance-heavy industries.

Common Use Cases

1. **Highly Secure Apps** – Apps requiring isolation from the public internet.
2. **Enterprise Applications** – Hosting critical business apps at scale.
3. **Regulatory Compliance** – Healthcare, Banking, Government workloads.
4. **Hybrid Connectivity** – Apps needing secure access to on-prem resources.
5. **Multi-tenant Applications** – SaaS providers hosting apps securely.
6. **Large-scale Hosting** – Thousands of apps in a single environment.
7. **Private APIs** – Hosting APIs only accessible within a private VNET.
8. **Disaster Recovery** – Running mission-critical apps in an isolated zone.

Best Practices

1. Use **ASEv3** instead of ASEv2 (simpler, more cost-efficient).
2. Place ASE in a **dedicated subnet** with proper NSG rules.
3. Use **internal ASE (ILB)** for private-only access.
4. Implement **WAF & Application Gateway** for internet-facing apps.
5. Use **Autoscaling** to handle demand fluctuations.
6. Integrate with **Azure Monitor + App Insights** for observability.
7. Secure outbound access with **firewalls or private endpoints**.
8. Choose the right **pricing tier (Isolated)** to balance cost and performance.
9. Deploy apps in **multiple AZs** for higher resiliency.
10. Use **DevOps automation** (ARM, Bicep, Terraform, pipelines) for deployment.

22. Azure Function

1. What is Azure Functions?
A **serverless compute service** that allows running code on-demand, triggered by events, without managing infrastructure.

2. What are triggers in Azure Functions?
Triggers **define how a function is invoked**, e.g., HTTP request, queue message, blob upload, timer.

3. What are bindings in Azure Functions?
Bindings simplify data integration → automatically connect inputs/outputs like Blob Storage, Cosmos DB, Service Bus without custom code.

4. What hosting plans are available for Functions?

- **Consumption Plan** – Pay per execution, auto-scale.
- **Premium Plan** – Pre-warmed instances, VNET integration.
- **Dedicated (App Service) Plan** – Reserved resources.

5. What is a Durable Function?
An extension of Azure Functions for **stateful, long-running workflows** (e.g., approval workflows, fan-out/fan-in processing).

6. How does Azure Functions scale?
Auto-scales horizontally → spins up more instances as event volume increases.

7. What is a cold start in Azure Functions?
Delay when a new instance is started after being idle → common in Consumption plan.

8. How can you reduce cold start impact?

- Use **Premium Plan** (pre-warmed instances).
- Keep function apps **warm with timer triggers**.

9. What languages are supported by Azure Functions?
C#, Java, Python, JavaScript, TypeScript, PowerShell, Go (preview).

10. How are functions deployed?

- Azure Portal
- Visual Studio / VS Code
- GitHub Actions / Azure DevOps CI/CD
- Zip Deploy / ARM / Terraform

11. How can you secure Azure Functions?

- Function Keys (default)
- Azure AD Authentication
- Managed Identity for service-to-service communication

12. How do you integrate Functions with other Azure services?
Using **triggers and bindings** (Service Bus, Event Hub, Blob, Cosmos DB).

13. Can Azure Functions run in a VNET?
Yes, but only in **Premium or Dedicated plan**, not Consumption.

14. How do you monitor Functions?

- **Azure Monitor + Application Insights** (logs, traces, metrics).

15. What's the max execution timeout for Functions?

- **Consumption Plan**: 5 min (can be extended to 10).
- **Premium/Dedicated Plan**: Unlimited.

16. When should you choose Azure Functions over Logic Apps?

- Use **Functions** for custom code.
- Use **Logic Apps** for workflow automation with connectors.

17. Can Functions be stateful?
Yes, using **Durable Functions**.

18. What's the difference between Azure Functions and Azure App Service?

- **App Service:** Long-running apps, web hosting, APIs.
- **Functions:** Short-running, event-driven, serverless compute.

19. How do you handle retries in Functions?

- **Queue/Service Bus triggers** automatically retry.
- Can configure **retry policies** in host.json.

20. How would you design a large-scale event-driven system with Functions?

- Use **Event Hub/Service Bus** for buffering.
- Deploy Functions in **Premium Plan** for performance.
- Use **Durable Functions** for orchestration.
- Monitor with **App Insights**.

Key Features

1. **Serverless Computing** – No need to manage infrastructure.
2. **Event-driven** – Triggered by events (HTTP requests, queues, timers, etc.).
3. **Multiple Triggers** – HTTP, Timer, Service Bus, Event Hub, Blob Storage, Cosmos DB.

4. **Bindings** – Simplify input/output (e.g., read from Blob and write to Cosmos DB).
5. **Scalability** – Auto-scales based on demand.
6. **Pay-per-Use** – Consumption plan charges only for execution time.
7. **Multiple Languages** – C#, Java, JavaScript, Python, PowerShell, Go, etc.
8. **Durable Functions** – Support long-running workflows and stateful orchestration.
9. **Security** – Authentication via Azure AD, API Keys, or Function-level access control.
10. **Integration** – Works seamlessly with other Azure services.

Common Use Cases

1. **Data Processing** – Process files, logs, or IoT device data.
2. **Real-time Notifications** – Trigger alerts based on events.
3. **API Backend** – Build lightweight APIs.
4. **IoT Processing** – Handle messages from IoT devices.
5. **Scheduled Jobs** – Automated tasks using Timer trigger.
6. **ETL Pipelines** – Ingest, transform, and load data.
7. **Image Processing** – Run transformations on uploaded images.
8. **Event-driven Workflows** – Process Service Bus/Event Hub messages.
9. **Chatbots & Webhooks** – Process bot/webhook messages.
10. **Microservices** – Build small, independent, serverless components.

Best Practices

1. Choose the **right hosting plan** (Consumption, Premium, Dedicated).
2. Use **Durable Functions** for workflows instead of chaining functions manually.
3. Keep functions **small and focused** (single responsibility).
4. Set **timeouts** (Consumption plan max = 5 min by default, extendable to 10).
5. Use **queues/topics** for retry and reliability.
6. Implement **logging & monitoring** with App Insights.
7. Secure functions with **Azure AD / Managed Identity**.
8. Optimize **cold starts** with Premium plan (pre-warmed instances).
9. Use **Deployment Slots** for zero-downtime deployments.
10. Apply **circuit-breaker/retry policies** for reliability.

23. Azure Logic App

1. What is Azure Logic Apps?
A **cloud-based integration and workflow automation service** that connects applications, data, and services using prebuilt connectors and a visual designer.

2. What is the difference between Logic Apps and Functions?

- **Logic Apps**: No-code/low-code workflow automation.
- **Functions**: Write custom code for event-driven tasks.

3. What are triggers in Logic Apps?
A **trigger** is an event that starts a workflow (e.g., new email, HTTP request, file upload).

4. What are actions in Logic Apps?
Steps that a workflow performs after a trigger (e.g., send email, write to SQL DB).

5. What is the difference between Stateful and Stateless workflows?

- **Stateful**: Keeps run history and state → useful for business processes.
- **Stateless**: Faster execution, no run history → useful for APIs.

6. How do Logic Apps scale?
They are **serverless** and automatically scale based on demand.

7. What is an Integration Account in Logic Apps?
A resource that supports **B2B/EDI scenarios** by storing schemas, maps, certificates, and agreements.

8. How does Logic Apps connect to on-premises systems?
Using the **On-Premises Data Gateway**.

9. Can Logic Apps run inside a VNET?
Yes, **Integration Service Environment (ISE)** allows running Logic Apps in a private VNET.

10. What is the difference between Standard and Consumption Logic Apps?

- **Consumption**: Pay-per-execution, multi-tenant, limited control.
- **Standard**: Runs in single-tenant, supports local development, better performance.

11. How do you secure Logic Apps?

- Managed Identity
- IP restrictions
- VNET Integration
- Private endpoints

12. How do you monitor Logic Apps?

- Azure Monitor
- Application Insights

- Run history in the portal

13. How do retries work in Logic Apps?
Most connectors support **built-in retries with exponential backoff**. You can configure retry count and interval.

14. Can Logic Apps call other Logic Apps?
Yes, using **"Call workflow"** action or HTTP actions.

15. What's the difference between Logic Apps and Power Automate?

- **Logic Apps**: Enterprise-grade integration, DevOps support.
- **Power Automate**: End-user automation with Office 365 focus.

16. When would you choose Logic Apps over Azure Functions?
When workflows require **multiple connectors and integrations** with minimal code.

17. Can Logic Apps process high-throughput data?
Yes, but for **low-latency, high-performance** use **Stateless workflows** or **Event Hub integration**.

18. How do you handle long-running workflows?
Use **Stateful Logic Apps**, which can run for **up to a year**.

19. How do you reduce costs in Logic Apps?

- Avoid frequent polling triggers (use push triggers).
- Combine steps where possible.
- Use child workflows to reuse logic.

20. Give an example of a real-world Logic App use case.
Example: An **invoice processing system** → Triggered when invoice PDF is uploaded to Blob, Logic App extracts data with AI Builder, stores in SQL DB, and sends approval email via Outlook.

Key Features

1. **Workflow Automation** – Automates workflows across services.
2. **No-Code/Low-Code** – Visual designer to build workflows.
3. **Large Connector Library** – 1,000+ connectors (Office 365, SAP, SQL, Service Bus, Salesforce, etc.).
4. **Triggers & Actions** – Workflows start with triggers and perform actions.
5. **Hybrid Integration** – Can connect to on-premises systems using Integration Runtime.
6. **B2B & EDI Support** – Supports protocols like AS2, X12, EDIFACT.
7. **Stateful and Stateless** – Stateful workflows store run history; stateless for low-latency high-performance.
8. **Serverless Scaling** – Auto-scales based on demand.
9. **Error Handling & Retries** – Built-in retry policies and exception handling.
10. **Integration with Azure Monitor & App Insights** – For logging and monitoring.

Common Use Cases

1. **Automated Email Processing** – Trigger workflows when emails arrive.
2. **Approval Workflows** – HR leave approval, document approval.
3. **Data Integration** – Sync between SQL, Salesforce, SharePoint, etc.
4. **IoT Data Processing** – Ingest IoT messages and route them to analytics systems.
5. **B2B Integration** – Exchange EDI/X12/AS2 messages.
6. **Social Media Monitoring** – Trigger alerts on Twitter mentions, posts.
7. **Incident Management** – Integrate with ServiceNow, PagerDuty, or Teams.
8. **Document Processing** – Extract and store PDFs, invoices, or forms.
9. **RPA Alternative** – Replace manual repetitive tasks with automated workflows.
10. **Microservices Orchestration** – Coordinate between APIs and services.

Best Practices

1. Use **Stateless workflows** for low-latency and **Stateful** when you need history.
2. Always implement **retry and error handling**.
3. Use **parallel branches** to improve performance.
4. Keep workflows **modular** by calling child workflows.
5. Use **Managed Identity** instead of credentials.
6. Apply **versioning** to avoid breaking changes.
7. Monitor workflows with **App Insights**.
8. Use **integration accounts** for B2B/EDI processing.
9. Optimize for **cost** by minimizing polling triggers.
10. Secure workflows with **VNET, IP filtering, and private endpoints**.

24. Azure Service Fabric

1. What is Azure Service Fabric?
A **distributed systems platform** for building and managing scalable, reliable microservices and container-based applications.

2. How is Service Fabric different from Kubernetes?

- **Service Fabric** supports **stateful microservices** and multiple programming models (actors, services).
- **Kubernetes** mainly orchestrates **stateless containers**.

3. What are Stateful and Stateless services in Service Fabric?

- **Stateless**: Don't maintain data between requests (e.g., web APIs).
- **Stateful**: Maintain state across requests, stored within the service itself.

4. What are Reliable Services in Service Fabric?
A programming model that provides APIs for building **stateful/stateless services** with reliability, partitioning, and replication.

5. What are Reliable Actors in Service Fabric?
An **actor-based model** where each actor is a single-threaded object that maintains its own state (good for IoT, gaming).

6. What is a Service Fabric Cluster?
A **set of virtual or physical machines** that together host and manage applications.

7. What are Service Fabric Partitions?
Partitions divide a service's workload or state across multiple nodes to achieve **scalability and reliability**.

8. How does Service Fabric ensure high availability?

- Replicates state across nodes.
- Performs **automatic failover** if nodes go down.

9. What are Service Fabric Application Types?
A **logical grouping of services** packaged and deployed together, reusable across clusters.

10. How do rolling upgrades work in Service Fabric?
Applications are upgraded **in phases**, moving traffic gradually, with **health checks** to prevent downtime.

11. How do you secure Service Fabric clusters?

- X.509 certificates for node-to-node and client communication.
- Azure AD authentication for management access.

12. How do you monitor Service Fabric applications?

- **Azure Monitor** and **App Insights** for telemetry.
- Built-in **health reporting** at service and cluster levels.

13. What are Health Reports in Service Fabric?
Reports from nodes, services, or infrastructure that determine whether the cluster and apps are healthy.

14. What disaster recovery options exist for Service Fabric?

- Geo-replication across regions.
- Backup and restore for stateful services.

15. How does Service Fabric handle scaling?

- Add/remove nodes in the cluster.
- Partition services for horizontal scaling.

16. When should you use Service Fabric instead of AKS?

- When you need **stateful microservices**.
- When you require **actors model** or mixed workloads (containers + .NET services).

17. What are common challenges with Service Fabric?

- Complex learning curve.
- Requires careful design for **partitioning and failover**.

18. Can Service Fabric run outside Azure?
Yes, it runs **on-premises**, in other clouds, or in Azure.

19. How does Service Fabric handle upgrades safely?

- Uses **rolling upgrades** with rollback on failure.
- Health policies ensure safe deployment.

20. Give a real-world example of Service Fabric usage.
Microsoft uses Service Fabric to run **Azure SQL Database, Cosmos DB, Cortana, and Skype for Business**, proving it's production-grade.

Key Features

1. **Microservices Platform** – Runs stateless and stateful microservices.
2. **Container Orchestration** – Supports Docker containers and Windows/Linux apps.
3. **High Availability** – Automatic failover and self-healing clusters.
4. **Scalability** – Elastic scaling of applications and services.
5. **Stateful Services** – Unlike Kubernetes, Service Fabric natively supports persistence.
6. **Rolling Upgrades** – Zero-downtime application and infrastructure upgrades.

7. **Multi-environment Support** – Runs on Azure, on-premises, or other clouds.
8. **Service Discovery** – Built-in naming and discovery service.
9. **Reliable Actors and Services** – Programming models for distributed systems.
10. **Monitoring & Diagnostics** – Integrated with Azure Monitor and App Insights.

Common Use Cases

1. **Microservices Applications** – Deploy large-scale distributed microservices.
2. **Financial Services** – Handle high-volume, stateful transaction processing.
3. **Gaming Platforms** – Manage multiplayer game sessions with low latency.
4. **IoT Platforms** – Collect and process device telemetry at scale.
5. **eCommerce Systems** – Cart, inventory, and order management.
6. **Container Hosting** – Run Docker containers alongside microservices.
7. **Workflow Engines** – Orchestrate long-running workflows.
8. **Telecom Systems** – Process millions of events per second.
9. **Multi-tenant SaaS Platforms** – Manage large numbers of customers.
10. **Hybrid Deployments** – Applications spanning on-prem and Azure.

Best Practices

1. **Choose the right programming model** (stateless, stateful, actors, containers).
2. **Partition stateful services** for scalability and reliability.
3. **Use rolling upgrades** with health checks for zero downtime.
4. **Design for failures** – always assume node failures will occur.
5. **Enable diagnostics & monitoring** (Azure Monitor, App Insights).
6. **Secure cluster communication** with X.509 certificates and Azure AD.
7. **Use managed Service Fabric clusters in Azure** to reduce operational overhead.
8. **Plan for backup & recovery** of stateful services.
9. **Optimize load distribution** with service partitioning.
10. **Consider AKS if only containers are needed** – Service Fabric is better for mixed workloads with state.

25. Azure API Management

1. What is Azure API Management?
A fully managed service that **publishes, secures, transforms, monitors, and manages APIs** in a centralized way.

2. Why do we need API Management?
- Centralized **API gateway**
- **Security** (OAuth, JWT, keys)
- **Traffic control** (rate limiting, quotas)
- **Monitoring & analytics**
- **Developer portal** for onboarding

3. What are the main components of APIM?
- **API Gateway** – Entry point for APIs.
- **Developer Portal** – Self-service API discovery.
- **Publisher Portal** – API publishing & configuration.
- **Management Plane** – Administer APIM via Azure portal, CLI, or ARM.

4. What is an API policy in APIM?
Policies are **rules/configurations** applied to APIs, such as:
- Caching
- Transformation (XML ↔ JSON)
- Logging
- Rate limiting

5. What are APIM SKUs?
- **Consumption** – Serverless, pay-per-call.
- **Developer** – For dev/test.
- **Basic** – Entry-level.
- **Standard** – Production.
- **Premium** – Enterprise, multi-region, VNET support.

6. How does APIM secure APIs?
- OAuth 2.0 & OpenID Connect
- Subscription keys
- Certificates
- Azure AD authentication

7. How can you rate-limit API usage?
By applying **policies** (e.g., limit 100 calls/minute per user).

8. What's the difference between subscription keys and OAuth tokens?
- **Subscription Key** → Simple access control, static.
- **OAuth Token** → Dynamic, identity-based access control.

9. How do you expose internal APIs securely using APIM?

- Use **VNET Integration**
- Deploy APIM in **Internal Mode**
- Restrict access with **firewall rules + policies**

10. Can APIM be used in multi-cloud or hybrid scenarios?
Yes. APIM gateway can manage APIs hosted in **Azure, AWS, GCP, or on-premises**.

11. How does APIM handle API versioning?
Supports:

- **Path-based** (/v1/orders)
- **Query string** (?version=1)
- **Header-based** (x-api-version: 1)

12. What's the difference between Revisions and Versions in APIM?

- **Revision** → Minor update to existing API (non-breaking).
- **Version** → Major breaking change, requires consumer migration.

13. How do you integrate APIM with CI/CD pipelines?
Using **ARM templates, Bicep, or Terraform** + Azure DevOps/GitHub Actions.

14. How does APIM support monitoring?

- Built-in analytics
- **Azure Monitor & App Insights** integration
- **Log-to-Event Hub/Storage** for SIEM tools

15. How can APIM transform API requests/responses?

- Convert **SOAP to REST**
- XML ↔ JSON
- Modify headers, query parameters, payload

16. When should you use APIM Consumption SKU?
For **serverless, lightweight workloads** where cost efficiency is critical.

17. When would you choose Premium SKU?
For **enterprise workloads** requiring:

- Multi-region deployments
- VNET integration
- Large-scale API usage

18. How do you handle API throttling in APIM?
By configuring **rate-limit** or **quota** policies at API or product level.

19. How do you expose APIs to external developers?
Through the **Developer Portal**, with:

- API documentation
- Self-service key generation
- API testing UI

20. How do you troubleshoot performance issues in APIM?

- Check **Azure Monitor metrics** (latency, request count).
- Review **policy configurations** (overuse of transformations).
- Use **Application Insights** for tracing.

Key Features

1. **API Gateway** – Acts as a single-entry point for APIs.
2. **Security & Authentication** – OAuth 2.0, JWT validation, certificates, and subscription keys.
3. **Traffic Management** – Rate limiting, quotas, and throttling.
4. **Developer Portal** – Self-service portal for API discovery, documentation, and testing.
5. **Transformation** – Modify requests/responses (e.g., XML ↔ JSON).
6. **Versioning & Revisions** – Manage multiple API versions and backward compatibility.
7. **Policies** – Rules for caching, logging, validation, and transformation.
8. **Analytics & Monitoring** – Built-in analytics and integration with Azure Monitor.
9. **Hybrid & multi-cloud support** – APIs can be hosted anywhere.
10. **DevOps integration** – Supports CI/CD and Infrastructure as Code (ARM, Bicep, Terraform).

Common Use Cases

1. **API Gateway** – Centralize access to backend APIs.
2. **Security Enforcement** – Protect APIs using OAuth 2.0, keys, or Azure AD.
3. **Multi-channel Applications** – Expose APIs for web, mobile, and IoT apps.
4. **Microservices Integration** – Aggregate microservices behind a single API gateway.
5. **Legacy Modernization** – Transform SOAP/XML APIs into REST/JSON.
6. **Third-party Integration** – Securely expose APIs to external partners.
7. **Rate Limiting** – Prevent backend overload from excessive requests.
8. **API Versioning** – Manage multiple API versions during migration.
9. **Compliance Logging** – Ensure secure API usage tracking.
10. **Hybrid/multi-cloud APIs** – Central API management across environments.

Best Practices

1. Use **products & subscriptions** to manage access for different consumers.
2. Apply **policies** for caching, transformation, and security.
3. Enforce **rate limiting & quotas** to protect backends.
4. Enable **logging & monitoring** with Azure Monitor & App Insights.
5. Use **versioning** when APIs evolve.
6. Leverage **developer portal** for self-service onboarding.

7. Secure APIs with **OAuth 2.0, certificates, or Azure AD RBAC**.
8. Use **DevOps pipelines** for APIM configuration deployment.
9. Deploy APIM in **Premium SKU** for multi-region HA.

   Periodically audit **API access logs** for security and compliance

26. Azure Service BUS

1. What is Azure Service Bus and why is it used?
It's a **fully managed enterprise message broker** that provides **asynchronous messaging** between decoupled applications or services.

2. What's the difference between Service Bus Queue and Topic?

- **Queue** → Point-to-point (one consumer per message).
- **Topic** → Publish-subscribe (multiple subscribers can receive a copy).

3. How is Service Bus different from Azure Storage Queue?

- **Service Bus Queue:** Advanced features (sessions, DLQ, transactions, duplicate detection).
- **Storage Queue:** Simple, cost-effective, no ordering/transactional support.

4. What are Message Sessions in Service Bus?
They provide **FIFO (First-In-First-Out)** ordering and correlation of related messages.

5. What is a Dead-Letter Queue (DLQ)?
A sub-queue that stores **messages that cannot be delivered or processed** (expired TTL, too many delivery attempts, filter mismatch).

6. What is Duplicate Detection in Service Bus?
Ensures a message with the same **MessageId** is only processed once within a detection window.

7. What is Auto-forwarding?
Automatically forwards messages from one queue/topic to another queue or subscription.

8. How does Scheduled Messaging work?
You can set a **ScheduledEnqueueTimeUtc** property so that a message is delivered in the future.

9. What are Partitioned Queues/Topics?
They distribute messages across **multiple brokers/partitions** for **higher throughput and availability**.

10. How do Transactions work in Service Bus?
Allows **atomic operations** across multiple messages (e.g., sending to a queue and completing another message in one transaction).

11. How is Service Bus secured?
- **Shared Access Signature (SAS) keys**
- **Azure AD RBAC**
- **Managed Identity** for apps

12. What happens when a message expires?
If TTL is exceeded, the message is moved to **Dead-Letter Queue**.

13. How does Service Bus ensure message ordering?
Through **Sessions** (FIFO guarantee).

14. How does Service Bus ensure reliability?
- Multiple **redundant brokers**
- **Geo-disaster recovery**
- **Partitioning** and **DLQ**

15. What's the difference between Peek-Lock and Receive-and-Delete modes?
- **Peek-Lock:** Two-step, safe (lock first, then complete). Prevents data loss.
- **Receive-and-Delete:** One-step, faster but risk of data loss.

16. When would you use Queues vs Topics?
- **Queue:** One consumer (e.g., order processing).
- **Topic:** Many consumers (e.g., billing, shipping, notification systems all get the same order event).

17. Can Service Bus integrate with Azure Functions?
Yes, using **Service Bus Trigger** in Azure Functions for event-driven processing.

18. How does Service Bus support event-driven architectures?
By using **Topics + Subscriptions**, multiple subscribers can react to the same event independently.

19. What are some common use cases of Service Bus in microservices?
- Communication between independent microservices.
- Decoupling API calls.
- Reliable order/event delivery.

20. How do you monitor and troubleshoot Service Bus?
- **Azure Monitor Metrics** (active messages, DLQ count, latency).

- **Application Insights** (message tracing).
- **Service Bus Explorer** (to test and debug queues/topics).

Key Features

1. **Asynchronous Messaging** – Decouples producers and consumers.
2. **Messaging Models** –
    - **Queues** → Point-to-point communication.
    - **Topics & Subscriptions** → Publish/Subscribe model.
3. **Message Sessions** – Maintain ordered delivery (FIFO).
4. **Dead-Letter Queue (DLQ)** – Stores undeliverable messages.
5. **Duplicate Detection** – Avoids reprocessing the same message.
6. **Scheduled Messages** – Delay delivery until a specific time.
7. **Transactions** – Group multiple operations atomically.
8. **Geo-disaster Recovery** – Paired namespaces for failover.
9. **Auto-forwarding** – Route messages from one queue/topic to another.
10. **Security** – SAS tokens, RBAC, and Managed Identity support.

Common Use Cases

1. **Order Processing Systems** – Ensure reliable and sequential order handling.
2. **E-commerce** – Inventory updates, billing, notifications.
3. **Banking & Finance** – Fraud detection, payment processing workflows.
4. **IoT Applications** – Device-to-cloud messaging.
5. **Microservices Communication** – Decouple services with asynchronous messaging.
6. **Event-driven Applications** – Multiple subscribers to the same event.
7. **Workflow Orchestration** – Long-running business processes.
8. **Load Leveling** – Smooth traffic spikes with queues.

Best Practices

1. Use **Topics** for broadcast (Pub-Sub), **Queues** for point-to-point.
2. Enable **Duplicate Detection** for idempotency.
3. Always monitor and handle **Dead-Letter Queues**.
4. Set **TTL (Time-to-Live)** to avoid stale messages.
5. Use **Sessions** when ordering is required.
6. Scale with **Partitioned Queues/Topics** for higher throughput.
7. Use **RBAC + Managed Identity** instead of connection strings.
8. Monitor with **Azure Monitor, Metrics, and Application Insights**.
9. Use **Auto-forwarding** to simplify routing across topics/queues.
10. Batch **send/receive operations** to reduce cost and improve throughput.

27. Azure Monitor

1. What is Azure Monitor?
A **comprehensive monitoring solution** that collects, analyzes, and acts on telemetry from Azure, on-prem, and hybrid environments.

2. What is the difference between Metrics and Logs in Azure Monitor?

- **Metrics**: Numeric data (CPU %, Memory, Requests/sec), near real-time.
- **Logs**: Detailed event/trace data, queryable via KQL.

3. What is Application Insights in Azure Monitor?
A feature that provides **APM (Application Performance Monitoring)**, including request tracing, dependency tracking, and user analytics.

4. What is Log Analytics Workspace?
A centralized **data repository** where Azure Monitor stores log data for analysis via KQL.

5. What is KQL (Kusto Query Language)?
A query language used in **Log Analytics** to analyze log and telemetry data.

6. How does Azure Monitor integrate with Azure resources?
It automatically collects telemetry from Azure resources and can be extended with agents or diagnostic settings.

7. How do alerts work in Azure Monitor?
Alerts can be set on **metrics, logs, or Application Insights data** and delivered via **action groups** (email, SMS, webhook, Logic Apps).

8. What are Action Groups?
Collections of notification preferences (emails, SMS, Teams, automation) that define **who gets alerted and how**.

9. Can Azure Monitor integrate with ITSM tools?
Yes, integrates with **ServiceNow, PagerDuty, OpsGenie, and custom webhooks**.

10. What is the role of Diagnostic Settings in Azure Monitor?
They define how and where logs/metrics are sent — to **Log Analytics, Event Hub, or Storage Account**.

11. How does Azure Monitor support security monitoring?
It integrates with **Microsoft Sentinel** and **Defender for Cloud** to detect security threats.

12. How do you ensure log data retention?
By configuring **retention policies** in Log Analytics workspaces and exporting to Storage for long-term compliance.

13. What is autoscale in Azure Monitor?
A feature that **automatically increases/decreases compute resources** based on monitoring metrics (e.g., CPU usage > 70%).

14. How does distributed tracing work in Application Insights?
It **tracks requests across multiple services** (e.g., Function → Service Bus → API → Database) to pinpoint latency issues.

15. What's the difference between Azure Monitor and Log Analytics?

- **Azure Monitor** = Umbrella service.
- **Log Analytics** = A feature within Azure Monitor for log queries and analysis.

16. How would you monitor an Azure VM with Azure Monitor?

- Enable **VM Insights**.
- Collect metrics (CPU, Memory).
- Send logs to Log Analytics.
- Set alerts (e.g., CPU > 80%).

17. How do you reduce costs in Azure Monitor?

- Filter logs to collect only needed data.
- Adjust log retention periods.
- Use sampling in Application Insights.

18. How do you troubleshoot app failures with Azure Monitor?

- Use **Application Insights** for failed requests.
- Query exceptions in **Log Analytics**.
- Check dependencies (SQL, Service Bus) with distributed tracing.

19. What's the difference between Azure Monitor and Azure Sentinel?

- **Azure Monitor**: Performance and health monitoring.
- **Sentinel**: SIEM/SOAR for security monitoring.

20. Give an example of a real-world use case.
A retail company uses **Azure Monitor + App Insights** to:

- Track API performance.
- Send alerts when checkout latency > 2 sec.
- Auto-scale VMs during high traffic.
- Forward logs to Sentinel for security analysis.

Key Features

1. **Unified Monitoring Platform** – Monitors applications, infrastructure, and networks.
2. **Metrics & Logs** – Collects numeric performance metrics and detailed log data.
3. **Application Insights** – End-to-end observability for applications.
4. **Log Analytics** – Centralized query and analysis of logs using Kusto Query Language (KQL).
5. **Smart Alerts** – Create rules for thresholds, anomalies, or log queries.
6. **Dashboards & Workbooks** – Custom visualization of monitoring data.
7. **Integration** – Connects with Logic Apps, Event Hub, Power BI, ServiceNow, etc.
8. **Autoscale Support** – Automatically scales resources based on metrics.
9. **Distributed Tracing** – Tracks requests across microservices.
10. **Security Monitoring** – Works with Defender for Cloud and Sentinel.

Common Use Cases

1. **Application Performance Monitoring** – Track latency, failures, and dependencies.
2. **Infrastructure Monitoring** – Monitor VMs, AKS, App Services, and databases.
3. **Log Analysis** – Query logs for troubleshooting incidents.
4. **Alerting & Notification** – Proactive alerts for downtime or performance degradation.
5. **Autoscaling** – Automatically add/remove VM instances based on metrics.
6. **User Behavior Analytics** – Track how users interact with apps.
7. **Microservices Observability** – Monitor Service Bus, Functions, Logic Apps, and AKS.
8. **Security Auditing** – Forward logs to Sentinel for security incident detection.
9. **Business Insights** – Use telemetry for usage and adoption metrics.
10. **Hybrid Cloud Monitoring** – Monitor on-prem and cloud systems together.

Best Practices

1. Define a **monitoring strategy** aligned with business SLAs.
2. Use **structured logging** for easier query and correlation.
3. Leverage **Application Insights for app-level telemetry**.
4. Enable **resource-specific metrics** for better granularity.
5. Use **Log Analytics workspaces** to centralize logs.
6. Configure **action groups** for multi-channel alerts (email, Teams, PagerDuty).
7. Avoid alert fatigue by **tuning thresholds and using dynamic alerts**.
8. Archive logs to **Azure Storage** for long-term retention.
9. Automate remediation using **Logic Apps or Functions** on alerts.
10. Continuously optimize dashboards & reports for stakeholders.

28. Azure Application Insights

1. What is Azure Application Insights?
An **Application Performance Monitoring (APM)** service in Azure Monitor that tracks app performance, failures, dependencies, and user behavior.

2. What type of applications can Application Insights monitor?

- Web apps (.NET, Java, Node.js, Python, PHP)
- Mobile apps (iOS, Android, Xamarin)
- Serverless (Functions)
- Containers (AKS, App Services)

3. How does Application Insights collect data?
Through **SDK instrumentation** (installed in code) and **Azure Monitor agents** for dependencies.

4. What is distributed tracing in Application Insights?
A feature that **tracks requests across services** (e.g., API → Service Bus → Function → Database) to find bottlenecks.

5. What's the difference between Azure Monitor and Application Insights?

- **Azure Monitor**: Umbrella service for all telemetry.
- **Application Insights**: Focused on **application-level monitoring**.

6. What kind of telemetry data does Application Insights collect?

- Requests, dependencies, exceptions, logs, custom events, performance counters, user sessions.

7. What are Availability Tests in Application Insights?
Synthetic tests that ping your app from multiple global locations to check uptime.

8. How does Application Insights integrate with Log Analytics?
Application Insights sends logs/telemetry to a **Log Analytics workspace**, queryable with **KQL**.

9. What is the role of Sampling in Application Insights?
Reduces telemetry volume by **collecting only a subset** of data, lowering cost.

10. How do you instrument an application for Application Insights?

- Install **SDK/agent** in your app.
- Configure telemetry collection in **Azure Portal or code**.

11. How do you monitor live applications with Application Insights?
Using **Live Metrics Stream** for real-time performance and request tracking.

12. How can Application Insights help with troubleshooting performance issues?

- Identify slow dependencies (SQL, API).
- Trace failed requests.
- Analyze exceptions stack trace.

13. How does Smart Detection work in Application Insights?

Uses **machine learning** to detect performance anomalies automatically.

14. How does Application Insights handle user session tracking?

Captures **page views, session duration, user flows, and retention**.

15. How can you set alerts in Application Insights?

- Based on **metrics** (e.g., response time > 3s).
- Based on **log queries**.
- Integrated with **action groups** (email, Teams, PagerDuty).

16. When should you use Application Insights vs Log Analytics?

- **App Insights**: Application-level (latency, failures, user flows).
- **Log Analytics**: Infrastructure & custom log queries.

17. Can Application Insights monitor on-prem apps?

Yes, as long as the app is instrumented with the SDK/agent.

18. How do you ensure telemetry security?

- Use **encryption in transit** (TLS).
- Restrict access with **RBAC**.
- Avoid logging **PII or secrets**.

19. How do you reduce Application Insights costs?

- Enable **sampling**.
- Tune **log retention**.
- Export old data to **Storage/Blob**.

20. Give a real-world example of Application Insights usage.

A banking app uses Application Insights to:

- Detect login failures.
- Track user journeys (loan application drop-off).
- Monitor SQL query latency.
- Run global availability tests for uptime

Key Features

1. **Application Performance Monitoring (APM)** – End-to-end monitoring of apps.
2. **Distributed Tracing** – Tracks requests across multiple services.
3. **Live Metrics Stream** – Real-time monitoring of app health and performance.
4. **Dependency Tracking** – Monitors SQL, Cosmos DB, Service Bus, APIs.
5. **Exception & Failure Tracking** – Captures errors, exceptions, and crash details.
6. **Custom Telemetry** – Developers can log custom events and metrics.
7. **Availability Tests** – Synthetic monitoring from multiple geographies.
8. **Smart Detection** – AI-based anomaly detection.
9. **Integration with Azure Monitor** – Works seamlessly with logs, metrics, and alerts.
10. **Multi-platform Support** – Works for .NET, Java, Node.js, Python, Mobile apps, and containers.

Common Use Cases

1. **Web App Monitoring** – Monitor performance of Azure App Service or on-prem web apps.
2. **API Monitoring** – Track API response times and failures.
3. **User Behavior Analytics** – Understand user journeys and sessions.
4. **Distributed System Debugging** – Trace requests across microservices.
5. **Real-time Monitoring** – Use Live Metrics for production workloads.
6. **Error Analysis** – Detect and analyze exceptions and crashes.
7. **Release Monitoring** – Validate app performance after deployments.
8. **Synthetic Monitoring** – Run availability tests from global regions.
9. **Business Insights** – Track KPIs like conversion rates, order completion, churn.
10. **Container & Serverless Apps** – Monitor Functions, AKS, App Services.

Best Practices

1. Enable **distributed tracing** for microservices.
2. Use **sampling** to reduce telemetry volume and cost.
3. Instrument both **server and client applications**.
4. Define **custom KPIs** (business + technical).
5. Always configure **alerts** for failures and latency.
6. Use **availability tests** for mission-critical apps.
7. Integrate with **Azure Monitor & Log Analytics** for deeper analysis.
8. Apply **role-based access control (RBAC)** for telemetry access.
9. Use **dashboards/workbooks** for visual insights.
10. Automate remediation workflows with **Logic Apps/Functions**.

29. Azure Log Analytics

1. What is Azure Log Analytics?
A cloud-based tool within **Azure Monitor** that collects and analyzes logs and telemetry from multiple sources using **KQL**.

2. What is a Log Analytics Workspace?
A **central repository** where logs are stored, queried, and analyzed.

3. How does Log Analytics differ from Application Insights?

- **Log Analytics**: Infrastructure + platform monitoring.
- **Application Insights**: Application performance monitoring (APM).

4. Which language is used to query Log Analytics?
Kusto Query Language (KQL).

5. What are the main sources of data for Log Analytics?

- Azure resources (VMs, networking, SQL, etc.)
- Application Insights
- On-prem servers via agents
- Security logs via Sentinel

6. How do you connect Azure resources to Log Analytics?
Enable **Diagnostic Settings** or install the **Azure Monitor agent**.

7. Can you query logs across multiple subscriptions?
Yes, Log Analytics supports **cross-resource queries**.

8. What is the retention period for Log Analytics data?
Default **30 days**, configurable up to **2 years** (with additional cost).

9. What are tables in Log Analytics?
Logs are stored in **tables** (e.g., Heartbeat, SecurityEvent, AppRequests).

10. What is the difference between Azure Monitor metrics and logs?

- **Metrics**: Numeric, near real-time (e.g., CPU %, memory).
- **Logs**: Detailed records (errors, events, traces).

11. How do you use KQL in Log Analytics?
Example query:

```
Heartbeat
| summarize Count = count() by Computer, bin(TimeGenerated, 1h)
```

This counts heartbeats per computer per hour.

12. How do you create alerts in Log Analytics?

- Write a KQL query.
- Save as an **alert rule**.
- Configure action groups (email, Teams, webhook, Logic Apps).

13. How can Log Analytics help in troubleshooting?

- Query errors across services.
- Correlate metrics (CPU spike with app crashes).
- Investigate security anomalies.

14. How does Log Analytics integrate with Microsoft Sentinel?

Sentinel uses **Log Analytics as its data store** for security logs.

15. How do you monitor hybrid environments with Log Analytics?

Install **Azure Monitor Agent** (AMA) or **Log Analytics Agent** on on-prem VMs.

16. When would you choose Log Analytics vs Event Hub?

- **Log Analytics**: Analysis, monitoring, dashboards.
- **Event Hub**: High-volume streaming integration.

17. How do you ensure log security?

- Use **encryption in transit and at rest**.
- Restrict access with **RBAC**.
- Avoid storing **PII or secrets**.

18. Can Log Analytics integrate with third-party SIEM tools?

Yes, via **Diagnostic settings + Event Hub export**.

19. How do you optimize Log Analytics cost?

- Enable **log sampling**.
- Filter out noisy categories.
- Reduce retention.
- Use **Capacity Reservations** for predictable workloads.

20. Give a real-world scenario where Log Analytics is used.

An e-commerce platform uses Log Analytics to:

- Collect logs from App Services, SQL, and AKS.
- Detect failed transactions.
- Correlate with security events in Sentinel.
- Trigger alerts for unusual login attempts.

Key Features

1. **Centralized Log Collection** – Collects logs from Azure resources, VMs, containers, and on-premises servers.
2. **Kusto Query Language (KQL)** – Powerful querying for logs and metrics.
3. **Integration with Azure Monitor** – Forms the backend for Application Insights, Monitor, Sentinel, etc.
4. **Cross-Resource Querying** – Query multiple resources across subscriptions.
5. **Custom Dashboards** – Build workbooks and visualizations.
6. **Alerts & Automation** – Trigger alerts and remediation workflows based on queries.
7. **Data Export** – Send logs to Event Hub, Storage, or third-party SIEM tools.
8. **Retention Policies** – Configurable retention (default 30 days, up to 2 years).
9. **Security Integration** – Works with Microsoft Sentinel for SIEM/SOAR.
10. **Scalability** – Handles terabytes of logs per day.

Common Use Cases

1. **Infrastructure Monitoring** – Collecting VM, container, and network logs.
2. **Security Analytics** – Feeding logs into Sentinel for threat detection.
3. **Application Debugging** – Query logs from App Insights + custom sources.
4. **Compliance & Auditing** – Retaining logs for governance.
5. **Performance Monitoring** – Analyze CPU, memory, request/response times.
6. **Cost Optimization** – Track resource usage patterns.
7. **Failure Investigation** – Debug crashes, errors, and service outages.
8. **Hybrid Monitoring** – Collect logs from both Azure and on-prem servers.
9. **Business Insights** – Analyze custom telemetry for KPIs.
10. **Automation** – Trigger Logic Apps or Functions from log queries.

Best Practices

1. Use **KQL queries** efficiently (filters first, summarize later).
2. Configure **data retention** based on compliance needs.
3. **Enable diagnostic settings** for all critical resources.
4. Export logs to **Storage/Event Hub** for archival or downstream systems.
5. Use **Log Analytics Workspaces per environment** (e.g., Dev, Test, Prod).
6. Apply **role-based access control (RBAC)** for query access.
7. Create **workbooks and dashboards** for stakeholders.
8. Use **scheduled alerts** for critical patterns (e.g., login failures).
9. Regularly **review ingestion costs** (filter noisy logs).
10. Integrate with **Sentinel, Monitor, and Application Insights** for full visibility.

30. Azure DevOps Board

1. What is Azure DevOps Boards?
A tool within Azure DevOps for **work tracking, Agile project management, and collaboration** using Scrum and Kanban methodologies.

2. What types of work items are available in Boards?

- Epics
- Features
- User Stories
- Tasks
- Bugs
  (Custom work items can also be created.)

3. What is the difference between a backlog and a board?

- **Backlog**: Ordered list of work items (Epics, Features, Stories).
- **Board**: Visual Kanban/Scrum board for tracking progress.

4. What is the purpose of iteration paths in DevOps Boards?
They define **sprints or timeframes** for planning and tracking work.

5. What is an area path?
Represents the **scope of work** (e.g., project, team, or feature area).

6. How does Azure Boards support Scrum?

- Sprint planning
- Backlogs & taskboards
- Velocity & burndown charts

7. How does Azure Boards support Kanban?

- Drag-and-drop work items across columns
- WIP (Work-In-Progress) limits
- Cumulative flow diagrams

8. What is the hierarchy of work items in Boards?
Epic → Feature → User Story → Task/Bug.

9. How are bugs tracked in Boards?
As a separate **Bug work item** (can be treated as a Task or Story depending on process template).

10. Can we customize Boards?
Yes, you can customize columns, swimlanes, work item states, and workflows.

11. How does Boards integrate with GitHub or Azure Repos?
Commits and pull requests can be **linked to work items** for traceability.

12. What reporting features does Azure Boards provide?
- Velocity charts
- Burndown/burnup charts
- Cumulative flow diagrams
- Custom dashboards

13. What is a query in Azure Boards?
A **search/filter mechanism** to retrieve work items based on conditions (e.g., all active bugs in sprint).

14. How can you manage dependencies between work items?
Using **work item links** (e.g., Parent-Child, Predecessor-Successor).

15. How does Boards support large organizations with multiple teams?
Through **area paths, team settings, and portfolio backlogs** for scaling Agile.

16. How do you ensure backlog grooming in Boards?
- Regular backlog refinement meetings
- Removing duplicate/outdated items
- Prioritizing features based on business needs

17. What are WIP limits and why are they important?
Work-In-Progress limits restrict the number of items in each column to **avoid bottlenecks**.

18. How do you track team velocity in Boards?
Via the **Velocity Chart**, which shows completed story points per sprint.

19. What are some best practices for using queries?
- Save frequent queries for dashboards.
- Use queries for sprint retrospectives (e.g., uncompleted tasks).
- Share queries with team members.

20. Give a real-world example of using Azure Boards.
A fintech company uses Azure Boards to:
- Track user stories and bugs across multiple teams.
- Run 2-week sprints with velocity charts.
- Link pull requests in Azure Repos to work items.
- Create dashboards for stakeholders to see release progress.

Key Features

1. **Work Item Tracking** – Manage user stories, tasks, bugs, and features.
2. **Agile Boards** – Kanban & Scrum boards for team collaboration.
3. **Backlogs** – Prioritize and organize product backlog items (PBIs).
4. **Dashboards & Reports** – Real-time progress tracking.
5. **Customization** – Create custom fields, workflows, and states.
6. **Integration with GitHub & Azure Repos** – Link commits and pull requests to work items.
7. **Query-based Work Item Search** – Powerful query builder to filter and analyze work items.
8. **Capacity Planning & Sprint Planning** – Assign workload based on team velocity.
9. **Extensions & Marketplace** – Enhance boards with plugins (e.g., Slack, Teams).
10. **Analytics & Insights** – Velocity charts, burndown, cumulative flow diagrams.

Common Use Cases

1. **Agile Project Management** – Scrum or Kanban teams.
2. **Sprint Planning** – Organizing user stories and tasks for iterations.
3. **Bug Tracking** – Tracking and resolving defects in applications.
4. **Product Backlog Management** – Prioritizing features and epics.
5. **Release Planning** – Mapping work items to releases.
6. **Cross-team Collaboration** – Multiple teams managing shared backlogs.
7. **DevOps Integration** – Linking code changes and CI/CD pipelines with work items.
8. **Business Stakeholder Tracking** – Dashboards for executives and PMs.
9. **Hybrid Project Tracking** – Combining Agile with traditional project management.
10. **Compliance & Audit** – Keeping history of work item changes for governance.

Best Practices

1. Use **separate projects** for different teams/products.
2. Define a **consistent work item hierarchy** (Epics → Features → Stories → Tasks).
3. Use **tags** for flexible categorization.
4. Keep **backlogs clean** (review old items regularly).
5. Apply **WIP limits** in Kanban boards to prevent bottlenecks.
6. Configure **queries and dashboards** for visibility.
7. Automate **linking of commits & PRs** to work items.
8. Encourage **daily updates** in Boards instead of status meetings.
9. Use **area paths** and **iteration paths** for structure.
10. Leverage **analytics views** for reporting velocity and predictability.

31. Azure Security Center (Defender)

1. What is Azure Security Center?

A **cloud-native security management system** that provides **threat protection, compliance management, and posture improvement** across Azure, on-prem, and multi-cloud.

2. What is the difference between Azure Security Center Free and Standard (Defender) tier?

- **Free tier**: Secure Score, recommendations, compliance checks.
- **Standard (Defender)**: Advanced threat protection for workloads (VMs, Storage, SQL, AKS).

3. What is Secure Score in Security Center?
A **numeric score** that reflects your security posture, based on implemented vs recommended practices.

4. How does Security Center help with regulatory compliance?
It provides a **compliance dashboard** that maps your security posture to standards like **ISO 27001, NIST, PCI-DSS**.

5. Can Security Center monitor non-Azure workloads?
Yes, it supports **AWS, GCP, and on-premises environments** via connectors and agents.

6. What is Just-in-Time (JIT) VM access?
A feature that **reduces VM attack surface** by allowing RDP/SSH access only on-demand and for limited time.

7. What are Adaptive Application Controls?
A way to **whitelist allowed applications** on VMs and block untrusted ones.

8. How does Security Center detect threats?
Using **Microsoft Threat Intelligence + behavioral analytics + ML** to detect brute-force, malware, SQL injection, and lateral movement.

9. What Defender Plans are available in Security Center?
- Defender for Servers
- Defender for SQL
- Defender for Storage
- Defender for Key Vault
- Defender for Kubernetes & Containers
- Defender for App Service

10. How does Security Center integrate with Sentinel?
Security alerts from Security Center can be sent to **Microsoft Sentinel** for SIEM/SOAR automation.

11. What is the role of the Log Analytics workspace in Security Center?
Security data and alerts are stored in **Log Analytics workspaces**, which allow KQL-based analysis.

12. How does Security Center support hybrid security?
By installing the **Azure Monitor agent** on on-prem or non-Azure machines.

13. How does Security Center handle vulnerability management?
Integrates with tools like **Qualys** to scan VMs for vulnerabilities and missing patches.

14. How does Security Center ensure data security?

- TLS encryption in transit
- At-rest encryption with Azure Storage
- RBAC for access control

15. How does Security Center support DevSecOps?

- Integrates with CI/CD pipelines (e.g., scan images in ACR, enforce security policies).
- Provides recommendations before deployment.

16. How would you use Security Center in a financial company?

- Enable Defender for SQL to protect customer data.
- Use compliance dashboard for PCI-DSS.
- Apply JIT access to critical VMs.

17. How do you reduce false positives in Security Center?

- Tune alert sensitivity.
- Use **automation rules**.
- Exclude non-relevant resources from policies.

18. What are quick wins to improve Secure Score?

- Enable MFA for all users.
- Apply NSGs and firewalls.
- Enable encryption at rest.
- Patch vulnerable VMs.

19. How do you integrate Security Center with workflows?

By triggering **Logic Apps or Azure Functions** based on alerts (e.g., auto-disable compromised accounts).

20. What is the difference between Azure Sentinel and Security Center?

- **Security Center (Defender for Cloud)**: Cloud **security posture management (CSPM)** + **workload protection (CWP)**.
- **Sentinel**: SIEM/SOAR tool for **log aggregation, threat hunting, and incident response**.

Key Features

1. **Unified Security Management** – Centralized view of Azure, on-prem, and multi-cloud security.
2. **Secure Score** – Security posture rating with recommendations.
3. **Threat Protection** – Detects threats across VMs, databases, storage, and containers.
4. **Vulnerability Assessment** – Integration with tools like Qualys for VM vulnerability scanning.
5. **Just-in-Time VM Access** – Restricts VM access to reduce attack surface.
6. **Adaptive Application Controls** – Whitelisting for apps to reduce risk.
7. **Regulatory Compliance Dashboard** – Maps posture against standards like ISO, NIST, PCI-DSS.

8. **Network Security Recommendations** – Identifies misconfigurations in firewalls, NSGs.
9. **Defender Plans** – Advanced protection for workloads (VMs, SQL, Storage, Kubernetes, Key Vault).
10. **Integration with Sentinel** – Extended SIEM/SOAR capabilities.

Common Use Cases

1. **Continuous Security Monitoring** – Monitor Azure, hybrid, and multi-cloud environments.
2. **Compliance Management** – Ensure adherence to industry standards.
3. **Threat Detection & Response** – Identify brute-force, malware, and suspicious activities.
4. **Vulnerability Management** – Detect unpatched systems and misconfigurations.
5. **Cloud Workload Protection** – Protect VMs, SQL, Storage, Containers, AKS.
6. **Identity Security** – Monitor suspicious sign-ins, privilege escalations.
7. **DevSecOps Integration** – Embed security checks into CI/CD pipelines.
8. **Hybrid Security** – Extend security monitoring to on-premises with agents.
9. **JIT Access Control** – Reduce RDP/SSH brute force risks.
10. **Security Automation** – Auto-remediation via Logic Apps or Sentinel playbooks.

Best Practices

1. Regularly review and improve your **Secure Score**.
2. Enable **Defender Plans** for all workloads (VMs, SQL, Storage, Kubernetes, etc.).
3. Configure **Just-in-Time access** for all VMs.
4. Use **regulatory compliance dashboard** for audits.
5. Continuously monitor with **alerts and automated responses**.
6. Deploy **adaptive application controls** to whitelist trusted apps.
7. Integrate Security Center with **Sentinel** for SIEM capabilities.
8. Apply **RBAC and least privilege access** for Security Center roles.
9. Automate security recommendations with **Policies & Blueprints**.
10. Regularly perform **vulnerability scans** and remediate findings.

32. Azure Key vault

1. What is Azure Key Vault?
A **cloud-based service** that securely stores and manages **secrets, keys, and certificates**.

2. What's the difference between secrets, keys, and certificates in Key Vault?

- **Secrets**: Strings (passwords, connection strings).
- **Keys**: Cryptographic keys (RSA, EC) for encryption/signing.
- **Certificates**: SSL/TLS certs managed and secured in the vault.

3. How does Azure Key Vault integrate with Azure AD?
Key Vault uses **Azure AD authentication** and RBAC for access control.

4. What is a Managed Identity in Key Vault?
A feature that allows **apps/services to authenticate without credentials** when accessing Key Vault.

5. How does Key Vault ensure high availability?
It uses **geo-replication** and is deployed in **highly available clusters**.

6. What's the difference between access policies and RBAC in Key Vault?

- **Access Policies**: Legacy method (per vault).
- **RBAC**: Newer, centralized model using Azure AD roles.

7. What is purge protection in Key Vault?
Prevents permanent deletion of secrets/keys even after soft delete.

8. How do you monitor and audit Key Vault activity?
By enabling **diagnostic logging** to Azure Monitor, Log Analytics, Event Hub, or Storage.

9. Can Key Vault restrict access by network?
Yes, via **firewall rules and private endpoints**.

10. What compliance standards does Key Vault support?
FIPS 140-2, PCI-DSS, HIPAA, ISO 27001, and more.

11. How do you access secrets from an Azure Function or App Service?

- Enable a **Managed Identity**.
- Grant Key Vault access.
- Reference secrets via **Key Vault references** in app settings.

12. Can Key Vault store credentials for third-party apps?
Yes, any API keys, DB passwords, or tokens can be stored.

13. What is Key Rotation in Key Vault?
Automatic or manual renewal of **keys, secrets, and certificates** to reduce risk.

14. How does Key Vault integrate with Azure DevOps pipelines?
By using the **Azure Key Vault task** to fetch secrets securely during pipeline execution.

15. Can you import your own keys into Key Vault?
Yes, with **Bring Your Own Key (BYOK)** functionality.

16. How do you prevent developers from hardcoding secrets?
By **storing secrets in Key Vault** and accessing them via references or SDKs.

17. How does Key Vault support data encryption at rest?
Azure services (Storage, SQL, Disk) can use **keys stored in Key Vault** for encryption.

18. How do you secure Key Vault itself?

- Enable **RBAC**.
- Restrict access with **firewall + private endpoints**.
- Enable **logging + alerts**.

19. What happens if Key Vault is deleted?
With **Soft Delete enabled**, it can be recovered within the retention period.

20. Real-world example of Key Vault usage?
In a healthcare app:

- Store **database connection strings** as secrets.
- Protect **TLS certificates** for App Services.
- Use **HSM keys** for encrypting patient records in SQL.
- Integrate with **DevOps pipelines** for secret injection.

Key Features

1. **Secrets Management** – Store sensitive information like connection strings, passwords, API keys.
2. **Key Management** – Manage cryptographic keys for encryption/decryption, signing, and key wrapping.
3. **Certificate Management** – Secure storage and lifecycle management of SSL/TLS certificates.
4. **Hardware Security Module (HSM) Support** – FIPS 140-2 Level 2 or 3 certified.
5. **RBAC and Access Policies** – Restrict access using Azure AD and roles.
6. **Logging & Monitoring** – Track access via Azure Monitor, Log Analytics, and Event Grid.
7. **Soft Delete & Purge Protection** – Prevent accidental deletion of secrets/keys.
8. **Managed Identities Integration** – Allow applications to authenticate without storing credentials.
9. **Versioning** – Store multiple versions of secrets and keys.
10. **Scalability & Availability** – Highly available with geo-replication.

Common Use Cases

1. **Application Secrets Storage** – Store DB connection strings, API keys securely.
2. **Encryption Keys Management** – Manage keys for Azure Storage, SQL, and custom apps.
3. **TLS/SSL Certificate Lifecycle** – Securely manage certificates for App Services, VMs.
4. **Compliance & Governance** – Enforce security policies (e.g., PCI-DSS, HIPAA).
5. **Secure DevOps Pipelines** – Integrate secrets in Azure DevOps & GitHub Actions.
6. **Key Rotation** – Automatically rotate secrets and certificates.
7. **Data Encryption at Rest** – Use keys stored in Key Vault with services like Azure Disk Encryption.
8. **Bring Your Own Key (BYOK)** – Import and manage customer-owned keys.
9. **Hybrid & Multi-Cloud Security** – Centralize secrets across multiple environments.

10. **Automated Secret Access** – Integrate with Managed Identity for serverless and microservices.

Best Practices

1. Always enable **Soft Delete and Purge Protection**.
2. Use **Managed Identities** instead of embedding credentials in code.
3. Enforce **least privilege** with RBAC and access policies.
4. Enable **logging** with Azure Monitor and send logs to SIEM.
5. Regularly **rotate keys and secrets**.
6. Avoid storing sensitive data in **app configs**; reference Key Vault instead.
7. Use **network restrictions (firewall + private endpoint)** for Key Vault.
8. Enable **HSM-protected keys** for compliance-sensitive workloads.
9. Organize secrets with clear **naming conventions**.
10. Regularly review and clean up unused versions of secrets.

33. Azure SQL database, Managed Instance

1. What is Azure SQL Database?

A fully managed **PaaS relational database** service based on SQL Server engine.

2. Difference between DTU and vCore pricing models?

- **DTU** = bundled CPU, memory, I/O.
- **vCore** = choose compute, memory, storage independently.

3. How does Azure SQL Database ensure high availability?

Via **automatic replication** across fault domains and zones with failover.

4. What is Hyperscale in Azure SQL Database?
A tier supporting **100TB+ storage** with distributed architecture.

5. What is Serverless SQL Database?
Auto-pauses when idle and resumes when activity occurs, charging only for used compute.

6. How does Point-in-time Restore work?
Automatic backups allow restoring to any second within the retention period (up to 35 days).

7. What is Elastic Pool?
A shared resource pool for multiple databases with unpredictable workloads.

8. How does Active Geo-replication work?
Up to 4 readable secondary replicas across regions for HA/DR.

9. Security features available?
TDE, Always Encrypted, Row-Level Security, Dynamic Data Masking, Firewall, Azure AD.

10. How to monitor performance?
Using **Query Store, Intelligent Insights, Automatic Tuning, Azure Monitor**.

11. Can SQL Database integrate with Power BI?
Yes, it's a direct data source for real-time reporting.

12. How does scaling work?
Scale up/down instantly via DTU or vCore.

13. Difference between Azure SQL Database and SQL Server on VM?

- SQL DB = PaaS, no infra management.
- SQL VM = IaaS, full control but more overhead.

14. How is disaster recovery implemented?
Active geo-replication or Auto-failover groups.

15. Authentication methods?
SQL logins, Azure AD authentication, Managed Identities.

16. How to handle multi-tenant SaaS workloads?
Use Elastic Pools or separate databases per tenant.

17. Can Azure SQL Database handle OLAP workloads?
Mostly OLTP, but can integrate with **Synapse Analytics** for OLAP.

18. What are Long-Term Retention backups?
Store backups in Azure Blob for up to 10 years.

19. How to secure network access?
Use Private Endpoint and disable public access.

20. Real-world scenario?
Retail platform: store customer orders, inventory in Azure SQL DB with elastic pools.

Key Features

1. **PaaS (Platform-as-a-Service)** – Fully managed database service.
2. **High Availability** – Built-in HA with 99.99% SLA.
3. **Scalability** – Scale up/down (DTU or vCore models).
4. **Automatic Backups** – PITR (Point-in-time restore) up to 35 days.
5. **Geo-replication** – Active geo-replication across regions.
6. **Advanced Security** – TDE, Always Encrypted, Row-Level Security.
7. **Monitoring** – Intelligent Insights, Query Performance tuning.
8. **Serverless + Hyperscale** – Auto-pause/resume and up to 100TB storage.
9. **Elastic Pools** – Share resources across multiple databases.
10. **Integration** – Works with Power BI, Logic Apps, Azure Functions.

Common Use Cases

1. **Modern cloud-native apps** needing a managed SQL backend.
2. **E-commerce systems** with elastic workloads.
3. **SaaS applications** serving multiple tenants.
4. **Data-driven reporting** integrated with Power BI.
5. **OLTP workloads** requiring high performance and resilience.

Best Practices

1. Choose **serverless or hyperscale** for variable workloads.
2. Use **elastic pools** for multiple low-usage DBs.
3. Enable **threat detection & auditing**.
4. Always configure **geo-replication for DR**.
5. Use **Private Endpoints** instead of public access.
6. Monitor performance with **Query Store**.
7. Enforce **Azure AD authentication** over SQL logins.
8. Enable **automatic tuning** for indexes.
9. Use **Long-Term Backup Retention (LTR)** for compliance.
10. Partition large tables for better performance.

1. What is Azure SQL Managed Instance?
A **fully managed PaaS SQL service** with near full SQL Server feature compatibility.

2. Difference between SQL DB and SQL MI?

- SQL DB = limited SQL Server features, internet accessible.
- SQL MI = almost full SQL Server feature set, deployed in VNet.

3. When would you prefer SQL MI over SQL DB?
When needing **SQL Agent, cross-database queries, SSIS, Service Broker**.

4. How is high availability provided in SQL MI?
By **Always On technology** behind the scenes.

5. Can SQL MI be deployed in a public network?
No, only inside a **VNet**.

6. How does SQL MI support migrations?
Via **Azure Database Migration Service (DMS)** or backup/restore.

7. What are auto-failover groups?
Feature to manage **multi-region failover** automatically.

8. What are networking requirements?
Deployed in **VNet with subnets**, supports VPN & ExpressRoute.

9. Does SQL MI support cross-database queries?
Yes, unlike Azure SQL Database.

10. Can SQL MI run SQL Server Agent jobs?
Yes, one of its major advantages.

11. What are some limitations of SQL MI?

- Higher cost than SQL DB.
- Deployment restricted to VNet.
- Startup provisioning time longer.

12. How does security differ vs SQL DB?
SQL MI always runs in a **private VNet**, offering better network isolation.

13. How are backups handled?
Automatic PITR backups, with long-term retention up to 10 years.

14. What about licensing?
Supports **Azure Hybrid Benefit** (reuse on-prem licenses).

15. What's the max storage size?
Up to **16 TB** per database.

16. Can SQL MI integrate with SSRS and SSIS?
Yes, supports both unlike SQL DB.

17. How does geo-replication work in SQL MI?
Using **auto-failover groups** across regions.

18. What are pricing tiers?

- **General Purpose**: standard workloads.
- **Business Critical**: high IOPS, low latency.

19. Can SQL MI handle OLAP workloads?
Better for OLTP; OLAP workloads should use **Synapse Analytics**.

20. Real-world example?
Bank migration: move from on-prem SQL to SQL MI with **SQL Agent jobs** and **cross-database joins** intact.

Key Features

1. **PaaS + Full SQL Server compatibility** – Supports SQL Agent, Service Broker, CLR.
2. **Lift-and-shift migration** – Minimal code change needed.
3. **Private VNet deployment** – More secure than SQL Database.
4. **High Availability** – Built-in with 99.99% SLA.
5. **Scalability** – Up to 16 TB storage, high vCore capacity.
6. **Automated backups** – PITR up to 35 days.
7. **Cross-database queries** – Unlike Azure SQL Database.
8. **Linked Server support** – Integrates with on-prem/other instances.
9. **Near 100% SQL Server feature parity** – Better for migration.
10. **Integration** – Works with SSRS, SSIS, third-party apps.

Common Use Cases

1. **Lift-and-shift migration** from on-prem SQL Server.
2. **Enterprise apps** requiring SQL Agent jobs, cross-database queries.
3. **Hybrid scenarios** – secure connectivity with VPN/ExpressRoute.
4. **Regulated industries** requiring private VNet deployments.
5. **Apps needing full SQL compatibility** but with managed infra.

Best Practices

1. Use **Auto-failover groups** for DR.
2. Enable **VNet integration** with NSGs.
3. Plan **storage and compute scaling** based on workload.
4. Use **Transparent Data Encryption (TDE)** and Always Encrypted.
5. Monitor with **Azure Monitor + Log Analytics**.
6. Configure **SQL Agent jobs** carefully (avoid high-frequency).
7. Regularly review **indexing & query tuning**.
8. Leverage **Hybrid Benefit** for licensing cost savings.
9. Enable **Geo-replication** for mission-critical workloads.
10. Separate **OLTP and reporting workloads**.

34. Azure Cosmos

1. What is Azure Cosmos DB?
A **fully managed, globally distributed, multi-model NoSQL database** designed for low latency and elastic scalability.

2. What models/APIs does Cosmos DB support?

- **SQL (Core API)** → JSON docs
- **MongoDB API** → Document DB workloads
- **Cassandra API** → Column-family
- **Gremlin API** → Graph
- **Table API** → Key-value

3. How does Cosmos DB achieve global distribution?
By replicating data across multiple Azure regions with **multi-master writes**.

4. What are Request Units (RUs)?
A **currency for throughput**; every operation (read/write/query) consumes RUs.

5. What are the consistency levels in Cosmos DB?

1. Strong
2. Bounded Staleness
3. Session
4. Consistent Prefix
5. Eventual

6. How does partitioning work in Cosmos DB?
Data is distributed across **logical partitions** (based on partition key) mapped to **physical partitions** for scaling.

7. What is Change Feed?
A log of changes (insert/update) in a container, used for **event-driven processing**.

8. How does multi-region write differ from single-region write?

- **Single-region write**: only one region accepts writes.
- **Multi-region write**: multiple regions can accept writes simultaneously.

9. How is high availability ensured?
With **replication across 4 replicas per region** and **99.999% availability SLA**.

10. What are the indexing policies in Cosmos DB?

- **Automatic indexing** by default.
- Can be customized (include/exclude paths, composite indexes).

11. How is data secured in Cosmos DB?

- Encryption at rest and in transit.
- RBAC & Azure AD integration.
- Private endpoints & firewalls.

12. How do you optimize RU consumption?

- Choose proper **partition key**.
- Use **stored procedures for batch ops**.
- Optimize queries and projections.
- Enable **autoscale RU**.

13. What is TTL (Time-to-Live) in Cosmos DB?
A setting to automatically delete items after a specified time.

14. How does Cosmos DB support disaster recovery?
Via **multi-region replication** and **automatic failover**.

15. How do you monitor performance and costs?
Using **Azure Monitor, Metrics, Application Insights, and Alerts**.

16. Cosmos DB vs Azure SQL Database – when to use which?

- **Cosmos DB** → NoSQL, high-scale, globally distributed, schema-free.
- **Azure SQL DB** → Relational, strong schema, OLTP workloads.

17. Example real-world use case of Cosmos DB?
A global **gaming app** where player data is replicated across regions for **low-latency access**.

18. Can Cosmos DB support schema changes easily?
Yes, since it's **schema-less (NoSQL)**.

19. How does Cosmos DB integrate with Azure Functions?
Functions can **trigger from Change Feed** or **read/write directly** to Cosmos DB.

20. What are some cost optimization strategies?

- Use **autoscale RU/s**.
- Apply **proper partitioning**.
- Set **TTL** for old data.
- Use **bulk operations**.

Key Features

1. **Globally Distributed** – Multi-region writes and reads with low latency (<10 ms).
2. **Multi-model Database** – Supports **document (SQL API), key-value (Table API), graph (Gremlin API), column-family (Cassandra API), and MongoDB API**.
3. **Elastic Scalability** – Scale throughput (RU/s) and storage independently.
4. **Guaranteed SLAs** – 99.999% availability, low latency, and consistency guarantees.
5. **5 Consistency Levels** – Strong, Bounded Staleness, Session, Consistent Prefix, Eventual.
6. **Partitioning** – Automatic horizontal scaling with logical partitions.

7. **Multi-master Support** – Write to multiple regions simultaneously.
8. **Change Feed** – Tracks inserts/updates in real-time for event-driven apps.
9. **Security** – Role-based access, private endpoints, encryption at rest & transit.
10. **Integration** – Works with Azure Functions, Synapse, Logic Apps, Power BI.

Common Use Cases

1. **IoT Applications** – Handling telemetry data at global scale.
2. **Real-time Retail Apps** – Shopping carts, order processing.
3. **Gaming** – Player data, leaderboards, matchmaking.
4. **Social Media Apps** – User profiles, messaging, activity feeds.
5. **Financial Services** – Fraud detection, transaction processing.
6. **Healthcare** – Patient data storage with HIPAA compliance.
7. **Global SaaS Applications** – Multi-tenant distributed workloads.
8. **Recommendation Engines** – Personalization and analytics.
9. **Supply Chain & Logistics** – Inventory tracking across geographies.
10. **Event-driven Systems** – Using Change Feed with Functions/Stream Analytics.

Best Practices

1. Choose the right **API** (SQL, Cassandra, MongoDB, Gremlin, Table) based on workload.
2. Select **partition key** carefully for even data distribution.
3. Use **Session consistency** for most real-world apps (balance between performance and correctness).
4. Enable **multi-region writes** for globally distributed apps.
5. Monitor and optimize **RU consumption**.
6. Use **Change Feed** for event sourcing and real-time triggers.
7. Secure Cosmos DB with **private endpoints and RBAC**.
8. Use **bulk executor library** for high-volume operations.
9. Implement **TTL (Time-to-Live)** for auto-deleting old data.
10. Regularly review **costs** and set **RU autoscaling** to avoid overprovisioning.

35. Azure Entra ID

1. **What is Entra ID?**
   A cloud-based identity and access management (IAM) service that provides authentication, authorization, and identity governance.

2. Difference between Azure AD (Entra ID) and on-prem AD?

- **AD**: Traditional, domain-joined, LDAP, Kerberos, used in LAN.
- **Entra ID**: Cloud-based, SAML/OAuth/OpenID, supports SaaS & cloud apps.

3. What protocols does Entra ID support?

- OAuth 2.0
- OpenID Connect
- SAML 2.0
- WS-Federation
- SCIM (for provisioning)

4. What is Single Sign-On (SSO)?
A feature that allows users to log in once and access multiple apps without reauthentication.

5. What's the difference between Entra ID Free, Premium P1, and P2?

- **Free** → Basic identity & SSO.
- **P1** → Conditional Access, Hybrid Identity, advanced security.
- **P2** → Identity Protection, Privileged Identity Management.

6. **What is Conditional Access in Entra ID?**
   Policies that control access based on **user, device, location, risk, or app sensitivity**.

7. What is Multi-Factor Authentication (MFA)?
An extra layer of security requiring **two or more verification methods** (password + phone/email/app).

8. What is Identity Protection in Entra ID?
AI-based detection of risky sign-ins, compromised accounts, and risk-based conditional access.

9. What is Privileged Identity Management (PIM)?
Provides **just-in-time access** for admin roles with approval workflows and auditing.

10. How does Entra ID support Zero Trust security?
By enforcing **MFA, Conditional Access, risk-based policies, and continuous evaluation**.

11. **Can Entra ID integrate with on-prem Active Directory?**
    Yes, using **Azure AD Connect** for hybrid identity.

12. **What is Azure AD Connect?**
    A sync tool that connects **on-prem AD with Entra ID** for hybrid user identity.

13. **What is B2B collaboration in Entra ID?**
    Allows external users (partners, vendors) to access apps/resources securely.

14. What is B2C in Entra ID?
A separate service that allows customer-facing apps to use social and enterprise identities for login.

15. How does Entra ID integrate with SaaS applications?
Via **SAML, OAuth, OIDC federation**, and pre-built connectors in Azure AD gallery.

16. How to secure privileged accounts?

- Enable MFA
- Use **PIM** for just-in-time access
- Restrict login locations with Conditional Access

17. What's the difference between App Registration and Enterprise Apps?

- **App Registration** → Used when you build/register a new app.
- **Enterprise App** → Pre-integrated or custom apps used in the tenant.

18. How does Entra ID handle device identity?

- **Azure AD Join** → Cloud-only.
- **Hybrid Join** → Linked to on-prem AD + Entra ID.
- **Registered** → Personal devices with limited access.

19. How can users reset their own passwords?
Using **Self-Service Password Reset (SSPR)**, if enabled by admin.

20. Real-world scenario of Entra ID usage?
A company using **Microsoft 365, Salesforce, and custom apps** can:

- Use **SSO** for seamless login.
- Enforce **MFA + Conditional Access** for security.
- Manage **guest partners** with B2B access.
- Protect **admins with PIM**.

Key Features

1. **Identity and Access Management (IAM)** – Centralized user & group management.
2. **Single Sign-On (SSO)** – Access multiple apps (cloud & on-prem) with one login.
3. **Multi-Factor Authentication (MFA)** – Strengthens authentication security.
4. **Conditional Access Policies** – Control access based on user, device, risk, location.
5. **Role-Based Access Control (RBAC)** – Assign least-privilege roles to resources.
6. **Self-Service Password Reset (SSPR)** – Reduce IT helpdesk dependency.
7. **Device Identity & Management** – Register and manage devices.
8. **B2B Collaboration** – Invite external users securely.
9. **B2C Identity Services** – Customer identity & access management.
10. **Monitoring & Governance** – Audit logs, Identity Protection, entitlement management.

Common Use Cases

1. **Centralized Authentication** – Secure login for Microsoft 365, Azure, and SaaS apps.
2. **Hybrid Identity** – Integrate with on-prem Active Directory.
3. **SSO for SaaS Apps** – Salesforce, ServiceNow, Google Workspace, etc.

4. **MFA & Conditional Access** – Enforce Zero Trust policies.
5. **B2B Collaboration** – Secure partner/vendor access.
6. **B2C Apps** – Provide customer logins with social identity providers (Google, Facebook).
7. **Identity Governance** – Role lifecycle, entitlement, and compliance.
8. **Privileged Identity Management (PIM)** – Just-in-time access for admins.
9. **Workforce Security** – Detect risky sign-ins and compromised accounts.
10. **Compliance & Auditing** – Meet ISO, HIPAA, and GDPR requirements.

Best Practices

1. Enable **MFA for all users** (especially admins).
2. Use **Conditional Access** with Zero Trust principles.
3. Prefer **Azure AD Join** + Intune for modern device management.
4. Enable **Identity Protection** for risk-based access control.
5. Apply **Privileged Identity Management (PIM)** for admin accounts.
6. Regularly review **sign-in logs and audit reports**.
7. Limit **guest access** with least privilege and expiration.
8. Use **Managed Identities** for apps instead of storing credentials.
9. Integrate with **SIEM solutions** (Sentinel, Splunk) for monitoring.
10. Educate users on **phishing-resistant authentication** (FIDO2, Passkeys).

Unlock the Power of Azure – From Beginner to Expert!

Are you preparing for Azure interviews? Or looking to master the skills that top companies demand in today's cloud-first world? This book is your ultimate roadmap.

THE ULTIMATE AZURE MASTERY (ZERO TO HERO) INTERVIEW GUIDE

is crafted to take you step by step through essential concepts, practical applications, and real-world scenarios that hiring managers love to ask. Whether you're just starting your Azure journey or already have hands-on experience, this guide is designed to give you the edge.

Inside you will discover:

- ✓ Core Azure concepts explained simply – from Virtual Machines to Databricks
- ✓ Peal interview-style Q&A – structured to mimic actual technical rounds
- ✓ Hands-on problem-solving strategies – bridging theory with practice
- ✓ Cloud architecture insights – learn how professionals design, optimize, and secure solutions on Azure
- ✓ Tips and tricks – to coniidently navigate tough interview questions

With clear explanations, carefully curated questions, and industry-relevant answers, this book transforms overwhelming Azure topics into a structured learning path.

Aaradhy Singh

is a cloud practitioner and eeducator passionate about helping aspiring professionals achieve their career goals, With a deep understanding of Azure and years of pracfical experience, Aaradhy blends technical accuracy with easy-to-foliow teaching.